**A picture containing text, screenshot, font

Description automatically generated**

**Master Thesis**

**Machine Learning and classification of transport mode choice using Python**

**Marios Melachroinos**

**Supervisor**

**Thanasis Argyriou**

**Athens 2024**

Contents

[2. Literature Review 2](#_Toc159508988)

[2.1. Factors influencing transport mode selection. 3](#_Toc159508989)

[2.2. Machine Learning essentials 4](#_Toc159508990)

[2.3. Classification Algorithms & Techniques 5](#_Toc159508991)

[2.4. Class imbalance and resampling strategies 10](#_Toc159508992)

[2.5. Machine Learning applications in transport mode choice 14](#_Toc159508993)

[3. Methodology 16](#_Toc159508994)

[3.1. Python Libraries 16](#_Toc159508995)

[3.2. Data collection for case 1: Thessaloniki 16](#_Toc159508996)

[4. Results 20](#_Toc159508997)

[4.1. Results for case 1 - Thessaloniki 20](#_Toc159508998)

[4.1.1. Data preparation and cleaning 20](#_Toc159508999)

[4.1.2. Explanatory Data Analysis 24](#_Toc159509000)

[4.1.3. Data Preprocess 37](#_Toc159509001)

[4.1.4. Decision Tree 42](#_Toc159509002)

[4.1.5. Random Forest 51](#_Toc159509003)

[4.1.6. XGBoost 59](#_Toc159509004)

[4.1.7. Stacked Model 72](#_Toc159509005)

[4.1.8. Model Selection 78](#_Toc159509006)

[References 83](#_Toc159509007)

# 2. Literature Review

Forecasting an individual's choice of transportation mode has garnered significant academic interest in recent years. The field of transportation and behavioural analysis has primarily relied on the extensive utilisation of logit models in existing research. In general, logit models and logistic regression analysis have been employed, specifically to examine the connection between the likelihood of binary or ordinal responses and explanatory variables using the maximum likelihood estimation method (Trueck, 2009). According to Trueck (2009), the logistic function is given by the following expression:

A black and white math equation

Description automatically generated

The above can also be rewritten as:

A math equation with a white background

Description automatically generated

McFadden (1972, 1974) posed a significant challenge to the initial approach to comprehending travel behaviour. In his exploration of the analysis of transit behaviour, he advocated for the application of the multinomial logit model. Like binary logit models, the multinomial model adheres to the same principles and assumptions, with the notable difference being the presence of multiple alternative choices (Lee & Kim, 2023). Furthermore, because of the assumption of independence among the options, the total probabilities of all choices add up to 1 (Lee & Kim, 2023). Following the introduction of the logit model, subsequent research endeavours have sought to extend this model and investigate travel behaviour, thereby addressing certain constraints associated with the original logit model. Such examples are the nested logit model (Willis, 2014) and the mixed logit model proposed by McFadden & Train (2000). The nested logit model organises choices into separate nests and permits varying correlations between these nests. As a result, correlations are consistent within each nest, but for options situated in different nests, the unobservable elements are uncorrelated and, in fact, entirely independent (Willis, 2014). The mixed logit model is defined as a standard multinomial model, also called “latent class model”, where the coefficients are chosen from a cumulative distribution, introducing an element of randomness (McFadden & Train, 2000).

## 2.1. Factors influencing transport mode selection.

Since the appearance of logit models, numerous studies have surfaced with the objective of delving into the crucial factors that affect transportation mode selection, going beyond the conventional determinants like cost or travel distance. In a study conducted by Mayo and Taboada in 2020, they employed a hierarchy model to assess the factors influencing the choice of public transportation mode among respondents in the Philippines. Their survey results indicated that safety was the most significant consideration, followed by cost, comfort, and concerns about environmental sustainability (Mayo & Taboada, 2020). Another study by Donkor et al. in 2020 examined the role of emotions in transport mode selection, focusing on respondents in the city of Edinburgh. Their findings revealed that an individual's feelings and experiences related to public transportation, along with their socio-demographic characteristics, exerted a substantial influence on their transit behaviour (Donkor et al., 2020). Additionally, McCarthy et al. in 2017 conducted further research suggesting that the presence of young children in a family had an impact on transportation behaviour. Specifically, they proposed that families with children preferred car usage over other sustainable transit options, with psychosocial factors and household characteristics playing pivotal roles in the choice of transportation mode (McCarthy et al., 2017). The influence of various weather conditions in transport mode choice have also been explored in the literature. Bocker et al. (2016) delved into weather-related factors and their connection to transit choices. Based on their analysis of travel diaries in the Netherlands, their results indicate that individuals who opt for walking or cycling modes are particularly affected by weather conditions (Bocker et al., 2016).

Additional research in the field of transportation focuses on mode selection during the COVID-19 pandemic and its impact on individual transit choices. To elaborate, Mussone & Changizi (2023) conducted a study that utilised a multinomial logistic regression model to investigate the factors influencing transportation mode choices prior, during, and post COVID-19 lockdowns. Their research, based on data from residents in Milan, Italy, indicated that socio-demographic factors, as well as individual preferences and concerns related to public transportation, played the most significant roles in predicting transport mode choices during the pandemic (Mussone & Changizi, 2023). It is worth noting that, despite mandatory contamination control measures, many residents in various countries expressed heightened concerns about the spread of the virus within public transportation during and after lockdown restrictions. Those concerns led residents to shift from relying on public transportation to utilising private vehicles for their commuting requirements. The phenomenon is further investigated by Das et al. (2021), who employed a logistic regression model to examine travel behaviour and modal transitions. Their research findings indicate that demographic factors exert a considerable influence on preferences for switching transportation modes. Additionally, trip-related factors, including travel time and health conditions, demonstrate a robust association with the inclination to shift from public transportation to using cars (Das et al., 2021).

## 2.2. Machine Learning essentials

An alternative to the traditional logit models, comes with the introduction of machine learning. According to Zhou (2021), Machine learning is a method that enhances the performance of systems through computational learning from prior experiences. In the realm of computer systems, these experiences are embodied in the form of data. The central objective of machine learning is to create learning algorithms capable of constructing models based on this data. When the learning algorithm is supplied with experiential data, it yields a model capable of making predictions for new observations (Zhou, 2021). Based on the presence or absence of labelled training data, learning problems can be categorised into two groups: supervised and unsupervised learning. Supervised learning encompasses a training phase in which the algorithm is supplied with a dataset comprising pairs of input and corresponding output (referred to as labelled data). During this phase, the algorithm acquires the ability to make predictions or classifications by drawing insights from this labelled data (Zhou, 2021). In supervised learning, the primary tasks are categorised into regression and classification, depending on whether the prediction output is continuous or discrete. In the case of classification problems, when there are only two possible labels, it is referred to as a "binary classification problem" (Zhou, 2021). If there are multiple possible labels, it is termed a "multi classification problem" (Zhou, 2021). Common algorithms that aim to solve regression and classification problems include Naïve Bayes Classifier, K-Nearest Neighbours, Decision Trees, Ensemble Learning, Boosting, Support Vector Machines and Neural Networks (Kubat, 2021).

In contrast, unsupervised learning is a category of machine learning in which the algorithm is given input data but does not have access to predefined output labels. In this context, the algorithm's role is to autonomously identify patterns, structures, or relationships within the data, all without prior knowledge of the expected output (Zhou, 2021). The main task in unsupervised learning is clustering which involves the process of categorising data points by identifying their similarities (Zhou, 2021). The most common technique for such problems is K-Means Clustering.

There is also a third form of learning called “Reinforcement learning” (Kubat, 2021). In this domain, the objective differs significantly. Here, the agent's role is not to induce knowledge from a pre-classified dataset but to engage in active experimentation with a system. The system, in turn, provides feedback in the form of rewards or penalties in response to the agent's actions. The agent's primary aim is to refine its behaviour by seeking to maximise rewards and minimise penalties as it interacts with the system (Kubat, 2021).

## 2.3. Classification Algorithms & Techniques

The most common algorithms and machine learning techniques that aim to solve classification tasks in a supervised problem are presented as followed:

***K-Nearest Neighbour (KNN)***

Perhaps the easiest and most straightforward algorithm for classification is that of KNN. The algorithm involves determining the class of a sample by identifying its closest neighbours and utilizing their characteristics. The term "k-Nearest Neighbour" is used because, in many cases, it is essential to consider more than just a single neighbour (k) to classify an example (Cunningham & Jane, 2021). Identifying the nearest neighbour is accomplished through the application of "Euclidean Distance" (Akanbi, 2014). The process in the algorithm, as well as the equation employed, are outlined as follows:

A black circle with a black dot

Description automatically generated

A black and white math symbol

Description automatically generated with medium confidence

Source: Akanbi & Fazeldehkordi, 2014

***Naïve Bayes Classifier (NBC)***

NBC is a machine learning algorithm that relies on the Bayes probability theory. Through the examination of the input data corresponding to a specified set of features, the Naïve Bayes classifier can calculate class probabilities associated with a specific label (Souza et al, 2021). To classify the input data, it is necessary to calculate the probabilities associated with each existing class. The class with the highest probability is then identified as the one to which the input data belongs (Souza et al, 2021). Thus, the classification task is to locate the class with the maximum probability to occur. The Bayes theory along with the max probability equation are presented below:

A black and white image of a mathematical equation

Description automatically generatedA black text on a white background

Description automatically generated

Source: Souza et al, 2021

***Decision Trees (DT)***

Decision trees serve as a versatile approach applicable to both regression and classification tasks. They stand out as one of the most employed algorithms, particularly suitable for classification tasks, offering numerous advantages over alternative classifiers. They are relatively easy to explain and interpret while requiring little effort for data preparation from the user (Benferhat & Elouedi, 2006). The structure of a typical decision tree can be observed in the figure below.

A diagram of a diagram

Description automatically generated

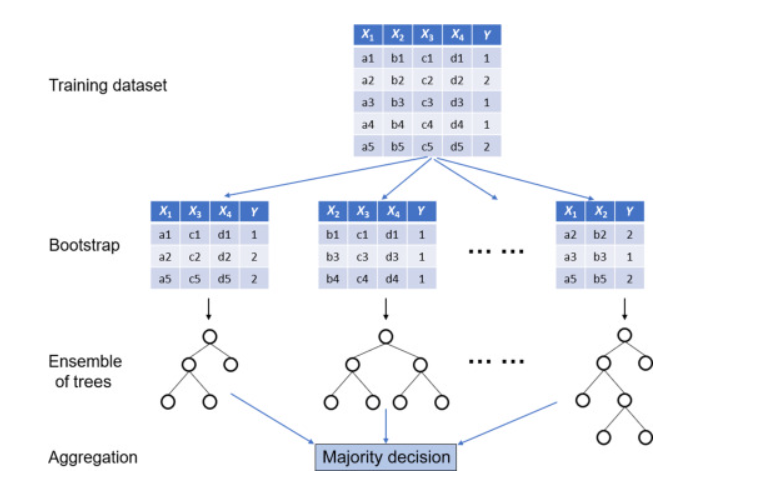
Source: Bellini, 2019

On a tree structure the root represents the starting point, the branches represent the splits/path of the root node, while decision nodes evaluate the features to split the data and which direction to follow.Lastly, the terminal nodes represent the outcome, or while in classification task, the class label (Bellini, 2019). A useful technique that is associated with decision trees is that of pruning, which is implemented in situations that the decision tree may grow too long so it is essential to reduce its decision nodes and avoid overfitting on the data (Bellini, 2019).

***Random Forest***

Frequently, it is crucial to construct more complex models by combining the predictions of several weaker models. This approach in data science is commonly referred to as the "ensemble" method. This procedure essentially combines different models that are trained to solve the same problem and get better results, making the final model more robust (Rocca, 2019).

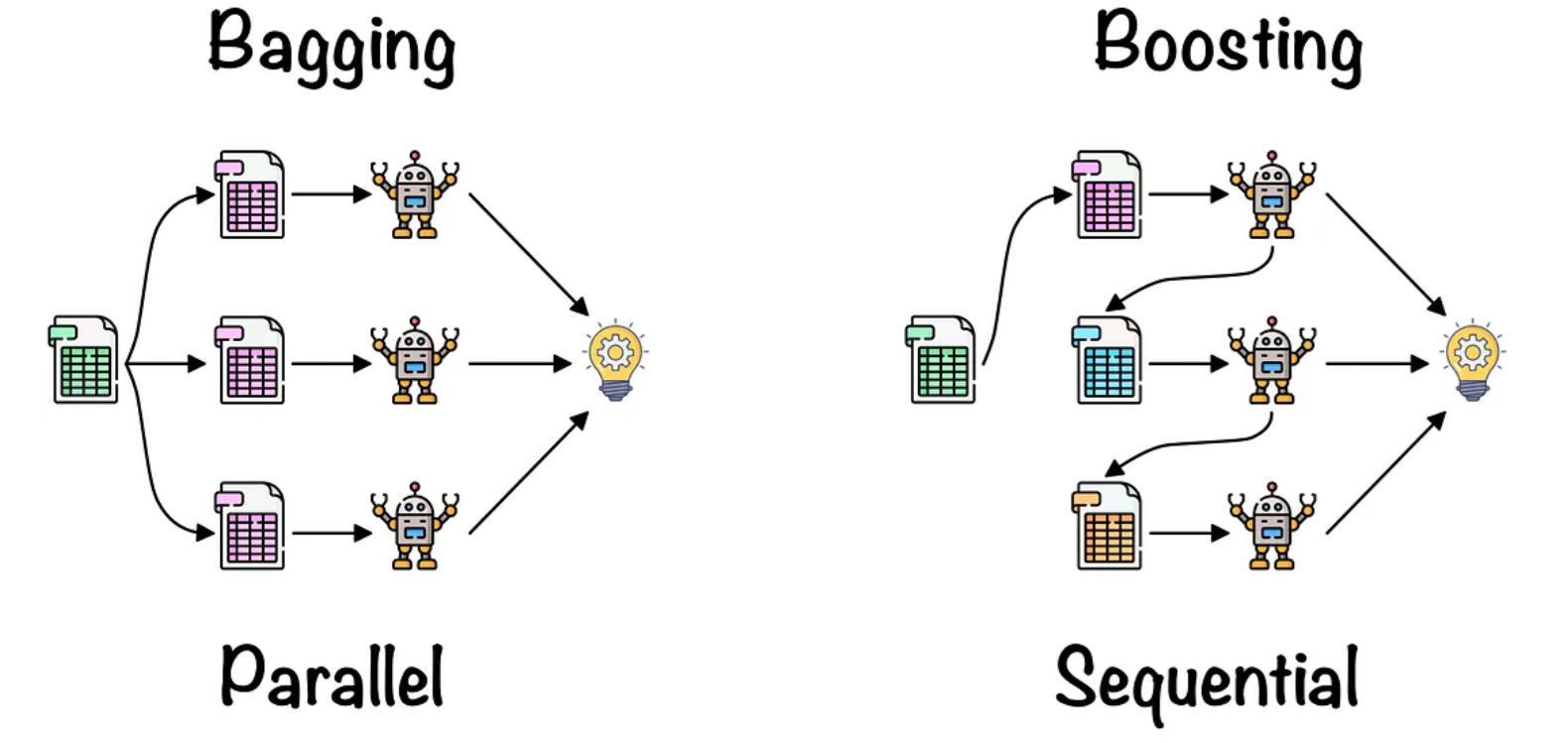
Random Forest stands out as the most widely adopted ensemble method for classification tasks, a model that trains multiple decision trees through a process called “bootstrapping”, followed by another process known as “bagging” aggregation. Bootstrapping refers to the training of individual trees in parallel and on different subsets of the training set, utilising a diverse set of features. This process ensures that each decision tree in the random forest is unique, which consequently reduces the variance of the classifier. For the model to make the final decision, the predictions from each individual tree are aggregated leading to a more generalised method. Due to its robust nature, the random forest often surpasses other classifiers in terms of accuracy. Despite being more complex in structure compared to an individual decision tree, the random forest is generally simpler when it comes to hyperparameter tuning (Misra & Li, 2020). A typical structure of the model can be viewed in the following image.



source: Misra & Li, 2020

***Boosting***

Another well-known ensemble method that combines numerous weak learners to create an enhanced model with improved accuracy and robustness. Comparable to the bagging technique of the random forest, boosting also involves aggregating predictions from each individual decision tree, resulting in a more generalised model. However, the primary distinction lies in the fact that, during boosting, the trees are trained sequentially, unlike the parallel training that takes place in the random forest. In this process, each model in the sequence is fitted with increased emphasis on observations in the dataset that were mishandled by the previous models. Essentially, each new model concentrates its efforts on the most challenging observations encountered so far. Consequently, by the end of the process, a robust classifier with reduced bias is obtained (Rocca, 2019). The process of boosting compared to bagging can be observed in the figure below.



Source: Lopez, 2021

Contrary to bagging and random forest, boosting exhibits numerous variations in the form of various algorithms. Such examples are:

* ***Adaptive Boosting***: The concept of adaptive boosting is to train multiple weak classifiers to create a robust classifier. In each iteration, the algorithm trains a weak classifier on a weighted version of the training data. The weight assigned to each training example is adjusted based on the previous weak classifiers. Adaptive boosting is implemented by the Adaboost Classifier algorithm (Polamuri, 2023).
* ***Gradient Boosting***: The process starts by training a weak learner using the initial data. It subsequently calculates the residuals, and another weak learner is fitted to the residuals. This process is repeated for a set number of iterations, where each new learner aims to minimise, the residuals left by the preceding learner. The model's prediction is the sum of all predictions from the weak learners (Polamuri, 2023).
* ***XGB (Extreme Gradient Boosting)***: An optimised version of gradient boosting, this technique has emerged as one of the most widely adopted methods for boosting procedures. This algorithm facilitates considerably faster training by enabling parallel processing and is adept at handling very large datasets with minimal requirements for data preprocessing. Like other boosting principles, XGBoost trains multiple week decision trees to create a strong model with increased accuracy and robustness (GeeksforGeeks, 2023).

***Stacking***

A third ensemble method is that of stacked generalization. In this approach multiple models are trained on the same dataset serving as the base estimators or “level-0 base models”. The predictions of those models are then used to train another model called as “level-1 meta model”. Hence, the output of the base models (predictions) is used as an input to the meta estimator. The main idea behind stacking generalization is to combine diverse model and combine the strengths of all those models. The meta estimator is then trained to combine those strengths for an enhanced predictive power (Brownlee, 2021).

The picture below illustrates the procedure of how stacking occurs. First the training set is split into folds. K-1 folds are used to train the base models, while the validation fold is used for predictions. The predictions from all the base models are then stacked and feed the meta learner who is responsible to make the final prediction.

A diagram of a training process

Description automatically generated

**Source:** [StackingCVClassifier: Stacking with cross-validation - mlxtend (rasbt.github.io)](https://rasbt.github.io/mlxtend/user_guide/classifier/StackingCVClassifier/)

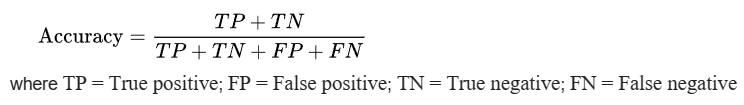
## 2.4. Class imbalance and resampling strategies

On many occasions, real world data comes with an issue called class imbalance. This occurs primarily on classification tasks when the distribution of the target variable is severely skewed, meaning that a certain class is underrepresented than others. Such situations occur often in disease detection and fraud transaction datasets where there are significantly fewer observations for diseased or fraud cases. This skewness often creates challenges for the models since they become more biassed towards the majority class, resulting in poor performance for the minority classes (Yang et al, 2024). To address this situation, resampling techniques are employed during the model training process, aiming to balance the distribution and minimise any bias in the model. The most widely used techniques are random **Undersampling** and **Oversampling**. In random undersampling, samples are deleted from the majority class until it matches the minority, though it can result in loss of information and underfitting. In contrast, with random oversampling, samples from the minority class are duplicated until it matches with the majority class. However, this considerably inflates the dataset size and increases the risk of overfitting in the training data. To mitigate such scenarios, methods like cross-validation, hyperparameter tuning, and regularisation are employed to minimise issues related to underfitting and overfitting. It is crucial though, that both of those techniques are only applied on the training set, since the test set serves as a genuine representation of the real-world problem and should always be left unaltered (Brownlee, 2021).

**Evaluation Metrics**

The ultimate objective of any classification model is to generate predictions for new and unseen data. Hence, to assess whether a built model is effective, various evaluation methods are necessary to make a concrete decision. Those metrics are presented as followed:

**Accuracy**

****

Typically, accuracy is defined as the ratio of correct predictions to the total number of instances. However, relying solely on accuracy as a metric can be overly optimistic, especially in situations with significant class imbalance. For instance, in a scenario with 1000 transactions, of which only 100 are fraud, a classifier could predict all cases as non-fraudulent, resulting in a 90% accuracy without detecting any of the fraud cases. To address this, additional metrics are considered that can emphasising on the evaluation of minority classes.

**Recall**

Also referred to as sensitivity, recall is a measure of how many truly relevant results are returned. It is defined as the number of true positives over the number of true positives plus the number of false negatives.

Recall is very used when you must correctly classify some event that has already occurred. It is frequently applied in scenarios such as fraud detection models or disease detection in patients. For instance, in the context of illness detection, it is crucial to identify individuals who are ill to prevent false negatives.

**Precision**

Precision is a measure of result relevancy. It is defined as the number of true positives over the number of true positives plus the number of false positives.



Precision is widely employed in marketing campaigns, particularly in the context of marketing automation. This is because a marketing automation campaign aims to initiate an activity for a user when the system predicts a successful response. In these scenarios, low precision translates to a financial loss as it entails reaching out to potential customers who are not interested in the marketing offer.

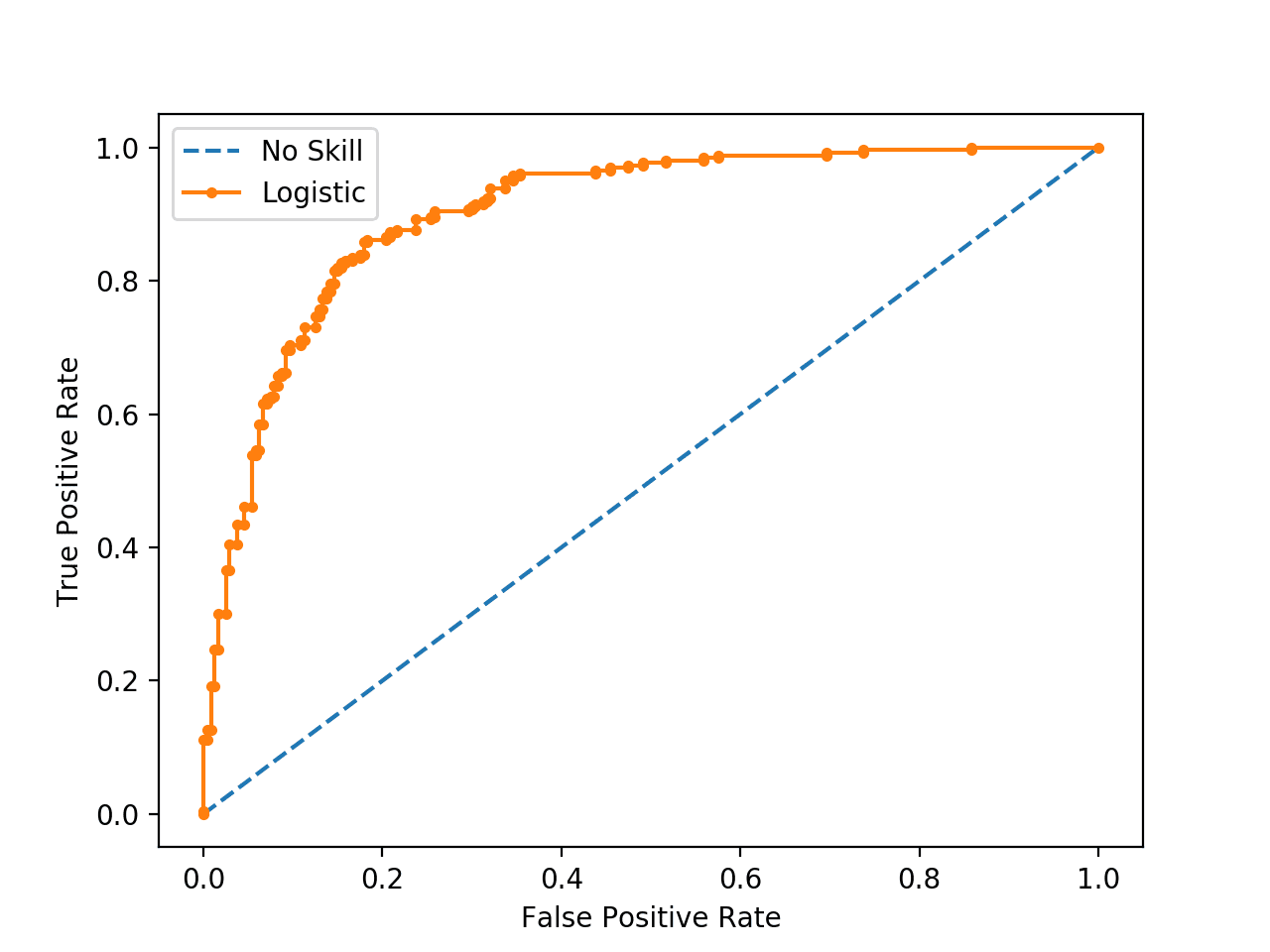
**F1 score**

F1 is the harmonic mean of precision and recall. It is often preferable in situations where there is class imbalance in the dataset, while both high recall and precision are desirable. The F1 score is expressed by the following equation. 

It can also be expressed in terms of precision and recall by the following expression.

**Receiver Operating Characteristic (ROC) Curve**

The ROC curve depicts the balance between the true positive rate (Sensitivity or Recall) and the false positive rate (FPR) across various probability thresholds. AUC (Area Under the Curve) is a metric assessing the classifier's ability to discriminate between the two. Its value ranges from 0 to 1, with values nearing 1 signifying optimal classifier performance. While commonly employed for binary classification tasks, it is possible to generate the curve for multiclass problems as well. In this context, each class has its individual curve, while Macro-AUC represents the average of the class-specific AUC values. An example of suck curve is depicted below.



Source: Brownlee, 2020

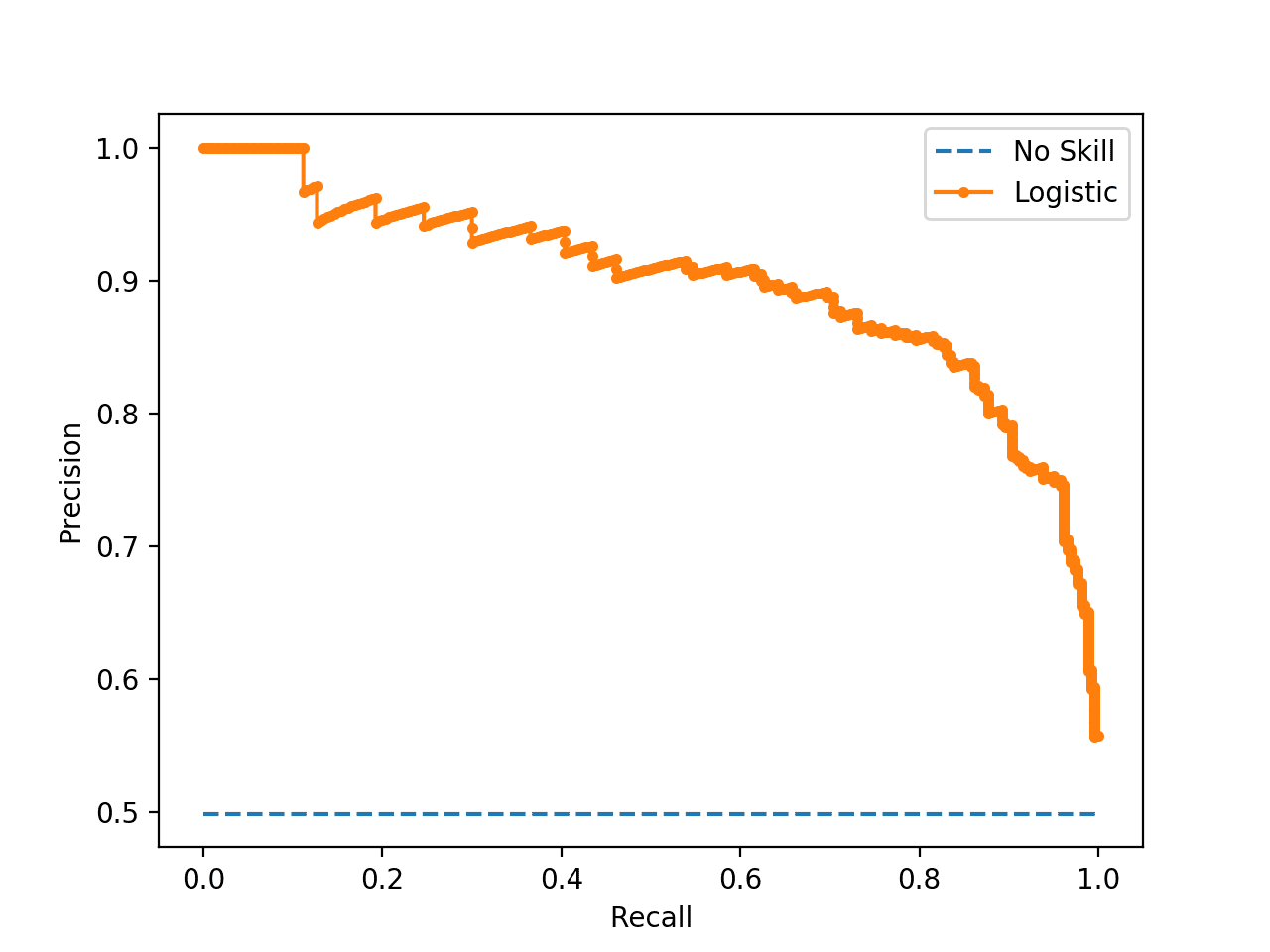
**Average Precision (AP)**

The relationship of recall and precision is depicted by the precision recall curve, which illustrates the trade-off between the two metrics for different probability thresholds. The area under the curve is known as Average Precision (AP) and is given by the following expression (Kashifi et al, 2022).



The area under the curve, like the ROC curve, has a value ranging from 0 to 1, with higher values indicating superior classifier performance. Precision-recall curves are preferred in scenarios with severe class imbalance since AUC in those occasions can be overly optimistic. Although commonly employed for binary classification, it is feasible to calculate curves for each individual class in a one-vs-rest (OvR) scenario. The micro-AP is then computed which is the harmonic mean of the class-specific AP values.

The pr curves are preferred in situations where there is a class imbalanced as they are more indicating for the performance of the minority classes. A typical pr curve is illustrated below.



Source: Brownlee, 2020

## 2.5. Machine Learning applications in transport mode choice

The adoption of machine learning in the modelling and prediction of transport mode choices has experienced a significant surge in recent times. In contrast to the prior logit models, the results indicate a notably improved ability to accurately predict transportation mode. This application is geared towards improving our understanding and predictive capabilities concerning individuals' decisions regarding the modes of transportation they choose. Numerous recent studies have sought to employ a range of machine learning models to predict transit behaviour. In 2015, Omrani conducted a study with the objective of forecasting the travel mode choices of individuals by applying machine learning techniques to national data from Luxembourg. His research outcomes revealed that artificial neural networks outperformed other alternative models in terms of predictive accuracy (Omrani, 2015). In their research conducted between 2010 and 2012 using data from the National Travel Survey in the Netherlands, Hagenauer and Helbich (2017) applied various machine learning classification models to predict travel mode choices effectively. Their results indicated that the Random Forest model outperformed the others. Nonetheless, they noted that while trip distance emerged as the most critical predictor, the importance of variables varied among different classifiers and class labels (Hagenauer & Helbich, 2017). In a parallel study using the same data, Kashifi et al (2022) set out to also forecast transportation mode choices. They extended the previous research by incorporating additional machine learning techniques into their analysis. Their results revealed that boosting and LightGBDT exhibited superior predictive accuracy for the different classes, especially when utilizing both under and oversampling methods to address class imbalance. Furthermore, their analysis underscored that age, income, and distance were the most influential predictors in the context of transport mode prediction (Kashifi et al., 2022).

Other applications of machine learning in the prediction of transportation modes can be characterised as more tailored to specific scenarios, as they are designed for modelling particular situations. In a recent study, Bhuiya et al (2022) focused on modelling transport mode choices for individuals with limited mobility in Dhaka. Their research findings indicated that multinomial logistic regression and linear discriminant analysis models exhibited superior predictive accuracy, particularly considering a smaller dataset (Bhuiya et al., 2022). Additionally, Zhao et al (2020) investigate differences between machine learning and logit models based on trip diary recordings. Their study findings from staff and students within the University of Michigan suggest that when deciding between machine learning and logit models for transport modelling, it seems there is a trade-off between predictive accuracy and the alignment with behavioural principles (Zhao et al, 2020).

Recent research developments in this field have introduced more sophisticated approaches, such as the adoption of artificial neural networks and deep learning methods. For instance, Zhang et al. (2020) introduced a deep neural network model for classification using data from Beijing, and their results demonstrate that this network model outperforms the random forest model in predicting transportation modes (Zhang et al., 2020). Furthermore, in a study based on national travel data from the UK, Bei et al. (2023) introduce a deep neural network model that goes beyond mere travel mode prediction, also addressing the purpose of the trip. Their research indicates that the model they proposed surpasses the performance of basic multinomial logit models and single-task neural networks (Bei et al., 2023). Additionally, Wang, Mo, and Zhao (2021) present a "theory-based residual neural network" model that integrates discrete choice models with basic neural networks, using three separate survey datasets. Their results indicate that the model they propose not only achieves superior predictive accuracy but also exhibits greater resilience compared to straightforward neural networks or discrete models (Wang, Mo & Zhao, 2021).

# 3. Methodology

## 3.1. Python Libraries

Below, the Python libraries utilised for the analysis dataset are displayed:

1. **Numpy**: It is one of the most implemented libraries, particularly in machine learning. This library is known for its support of matrices and multi-dimensional data. Numpy incorporates mathematical functions, among which the Array Interface that stands out as one of its most valuable features.
2. **Pandas**: It stands as a crucial library for data scientists, serving as an open-source machine learning library offering top notch analytical tools. It includes features such as sorting, visualisations, conversions and more.
3. **Matplotlib**: It is employed for visualising numerical data, making it a valuable tool in data analysis. It includes many unique charts such as pie charts, histograms, scatter plots and more.
4. **Scikit-learn**: A machine learning library that supports most supervised and unsupervised algorithms. The library works for many separate problems including regression, classification, clustering and more.
5. **Geopy**: A python library that enables users’ identification of coordinates for addresses, cities, countries internationally. It is also used for calculating distances between different coordinates (pypi.org, 2024)
6. **Seaborn**: It is used for data visualisation, offering a high-level interface to create visually appealing and informative statistical graphics. The library is built on top of the library of Matplotlib also functioning effectively with Pandas.
7. **Imbalanced-learn**: It is a library that offers various re-sampling techniques specifically designed for datasets with pronounced class imbalance. It is compatible with scikit-learn.

## 3.2. Data collection for case 1: Thessaloniki

Data for Thessaloniki were collected through an online survey using the Google Docs platform. While the survey was initially created in Greek, the analysis was conducted in English. The survey was distributed purely online through social media, emails and smartphones to residents in Thessaloniki. The following comprises the complete list of questions posed both in Greek and English

1. Διεύθυνση Κατοικίας - Home Address
2. Διεύθυνση Εργασίας - Work Address
3. Φύλλο - Gender
4. Ηλικία - Age
5. Έχετε δίπλωμα για οποιαδήποτε από τα παρακάτω: Αυτοκινητο, Μηχανη, Φορτηγό, Τίποτα - Do you have a license for any of the following: Car, Motorcycle, Truck, Nothing
6. Διαθέτετε κάποιο από τα παρακάτω: Ποδήλατο, πατίνι, τίποτα - Do you have any of the following: Bicycle, skates, nothing
7. Από πόσα άτομα αποτελείται η οικογένειά σας - How many people does your family consist of ?
8. Πόσα ιδιωτικά οχήματα διαθέτετε στην οικογένειά σας - How many private vehicles do you have in your family?
9. Μηνιαίο εισόδημα - Monthly Income
10. **Με ποιον τρόπο πηγαίνετε συνήθως στην εργασία σας - Mode for commuting to work (target variable)**
11. Αναφέρετε πόσα λεπτά στο περίπου χρειάζεται για να μεταβείτε από το σπίτι σας προς τον χώρο εργασίας σας - Indicate how many minutes approximately it takes you to travel from your home to your workplace.
12. Ποιές ώρες πηγαίνετε συνήθως στην εργασία σας - What time do you usually go to work?

Likert Based questions

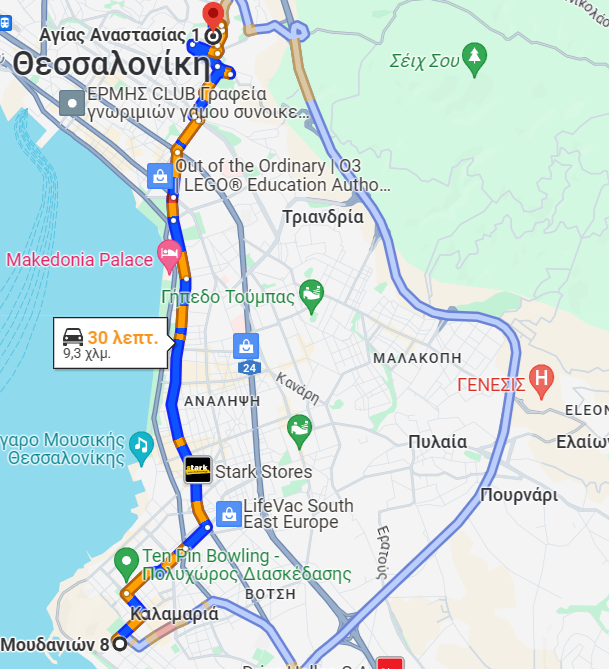
Do you think the following factors affect your commute?

Use scale where:

1-Strongly disagree , 2-Disagree , 3-Neutral , 4-Agree , 5-Strongly agree

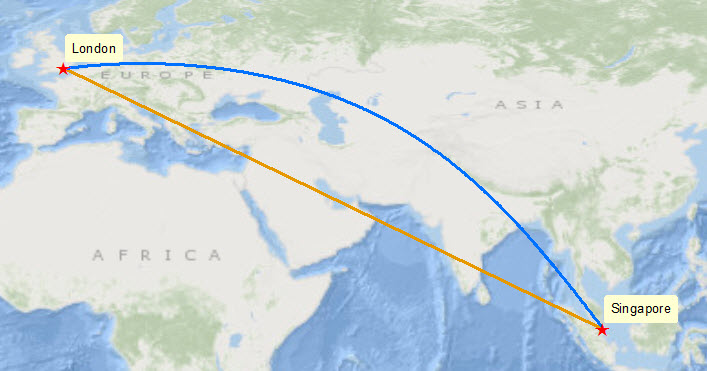
1. Άνεση Μετακίνησης - Convenience
2. Κόστος Μετακίνησης - Cost
3. Ασφάλεια μετακίνησης - Safety
4. Προστασία του περιβάλλοντος - Environmental concerns
5. Σωματική άσκηση και υγεία - Physical exercise and health
6. Καιρικές συνθήκες - Weather conditions
7. Διαθεσιμότητα χώρου στάθμευσης - Parking availability

The survey remained accessible from November 15 to January 15. A total of 409 samples were gathered, and after eliminating blank or erroneous responses as well as samples with unclear home or work addresses, 383 were deemed valid for subsequent analysis. Each question of the survey corresponds to a feature in the resulting dataset. Additional columns were generated for checkbox questions like 5, indicating binary values of either yes or no. For example, the possible answers in question 5 are Car, Motorcycle, Truck or Nothing. Three columns were created: Car\_licence, Motor\_licence and Truck\_licence that store whether the respondent has a licence for those or not. The same procedure was implemented also for question 6, creating two new features: Bike\_access and Skate\_access. Additionally, two distance-related features were generated using the Home address and Work address. More specifically, distance in kilometres was computed between home and work address using Google Maps. An illustration of the procedure is presented in the figure below:



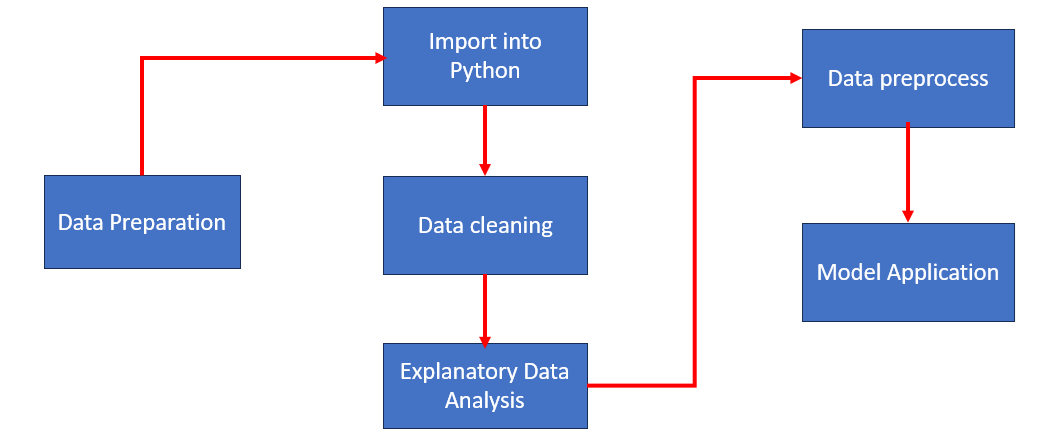
For samples where the mode of transportation was a car, the optimal distance for cars was chosen. If walking or taking the bus were the chosen modes of transport, the respective optimal distances were selected. Since optimal distances vary depending on the calculation time, the distances were computed during peak working hours: 06:00 - 09:00 and 14:00 - 17:00. Also, the corresponding time for each distance was used to validate the responses from the survey in question 11. Specifically, respondents were requested to indicate the duration of their commute to work in minutes. In cases where there was a substantial difference between their provided answers and the calculated time in Google Maps, the corresponding time from Google Maps was adopted. For example, if a respondent stated a 40-minute commute, but Google Maps suggested 25 minutes, the Maps estimate was used. In contrast, if a respondent implied a 30-minute commute and Google Maps indicated 33 or 34 minutes, the respondent's provided time was selected. This procedure was manually executed for all 383 samples.

The second distance-related feature was calculated in a more automated way using the Python library of Geopy. The library provides access in geocoding enabling users to find the geographic coordinates of a location based on the provided address. Based on those coordinates Geopy can calculate the distance between home and work address in a completely automated way. Though, the main difference from Google Maps, is that the corresponding distance is a geodesic one. Geodesic distance is the shortest distance between two points but on a curved surface like that of the earth (Kettle, 2014). This can be better understood for distances using the airplane, where in a 2D map the shortest path between two points seems to be a straight line, but the actual shortest path is the geodesic one because of the curved surface of the earth. This example is demonstrated in the figure below with the geodesic distance represented by the blue line.



**Source: Kettle, S., 2014**

The purpose of calculating two types of distances is to use them individually as features in classification models and evaluate whether there are significant differences in the model outcomes depending on the use of separate distance metrics. The resulting dataset comprises 383 samples and 24 features. The analysis procedure that was followed for the dataset is illustrated in the figure below:



Four Models will be used in total: Decision Tree, Random Forest, XGBoost and a Stacked model. The evaluation metrics include Precision, Recall, Accuracy and AUC scores from the ROC curves.

# 4. Results

## 4.1. Results for case 1 - Thessaloniki

This section of the thesis will present the findings related to Thessaloniki and commuting behaviour. The initial part covers data preparation and cleaning, followed by exploratory data analysis and data preprocessing. The final segment focuses on model application, mode classification, and the selection of the best model.

### 4.1.1. Data preparation and cleaning

Out of the 409 samples obtained from the online survey, blank responses were excluded initially. Subsequently, responses with errors in home or work addresses (those unidentifiable in Google Maps) were eliminated. Lastly, a few samples indicated respondents working from home or concealing either of their addresses; these samples were also excluded. Consequently, from the initial 409 samples, 383 were deemed valid. Furthermore, as detailed in the methodology section, additional features were generated for the two checkbox questions, considering only "yes" or "no" values. Following the creation of these features, the data were imported into Python for further preparation and cleaning. Below are the columns of the dataset as imported into the virtual environment:

The column names, excluding those derived from checkboxes, essentially correspond to the questions posed in the online survey. For better management during the analysis phase, it was necessary to rename the columns. The features used are illustrated on the table below.

|  |  |  |
| --- | --- | --- |
| **Columns** | **Type** | **Description** |
| Home\_address | object | Home address of the respondent |
| Work\_address | object | Work address of the respondent |
| Gender | Categorical-Binary | Gender of the respondent |
| Age | Categorical | Age group of the respondent |
| Driver\_licence | Categorical - Binary | Access to a driver licence? |
| Motor\_licence | Categorical - Binary | Access to a motor licence? |
| Car\_access | Categorical - Binary | Access to a car? |
| Bike\_access | Categorical - Binary | Access to a driver bike? |
| Motor\_access | Categorical - Binary | Access to a motorcycle? |
| Skate\_access | Categorical - Binary | Access to a skate? |
| Hsize | Numeric - Discrete | Household size |
| Vehicles | Numeric - Discrete | Number of private vehicles in household |
| Income | Categorical | Income group of respondents |
| Mode | Categorical | Mode for commuting to work  (Private vehicle, bus, walk) |
| Time | Numeric - Continuous | Minutes required for transit to work |
| Depart\_time | Categorical | Time of departure for commuting to work |
| Convenience | Categorical - Ordinal | Degree of influence of convenience (1 to 5) |
| Cost | Categorical - Ordinal | Degree of influence of cost (1 to 5) |
| Safety | Categorical - Ordinal | Degree of influence of safety (1 to 5) |
| Environment | Categorical - Ordinal | Degree of influence of environmental concerns (1 to 5) |
| Health | Categorical - Ordinal | Degree of influence of exercise and health (1 to 5) |
| Weather | Categorical - Ordinal | Degree of influence of weather (1 to 5) |
| Parking | Categorical - Ordinal | Degree of influence of parking availability (1 to 5) |
| Distance | Numeric - Continuous | Travel distance (kms) between home and work address |

In total, there are 24 features, comprising 4 numeric, 18 categorical, and 2 object features. The next steps involved checking for null values using the **df.isnull().any()** command and identifying duplicates with **df.duplicated().sum()**. Given that blank samples were eliminated prior to importing into Python, there were no instances of nulls or duplicates.

Furthermore, erroneous samples were also removed. There were cases where respondents indicated commuting to work by private vehicle without possessing a licence for either of those. Also, it was examined whether there are instances of respondents commuting to work with a private vehicle without having any in their household. This scenario was checked within a graph presented below.

A graph of different colored bars

Description automatically generated

It is identified that there are no erroneous instances for that scenario (Green bar). Afterwards, a lone sample was excluded, where the respondent suggested a trip distance exceeding 80 kilometres, constituting an outlier within the dataset. The last step was to calculate the geodesic distance using the geopy library in python. The code is illustrated below.

A screenshot of a computer code

Description automatically generated

There were samples where the process could not identify distances between addresses, so NaN values returned for those cases as shown below.

A screenshot of a cell phone

Description automatically generated

Those NaN values were replaced with the corresponding Distance (that was calculated manually). For instance, the above NaN value was replaced with 0.35. The same procedure was followed for all instances with NaN values. Consequently, 381 valid options were retained to proceed with the exploratory data analysis phase. The final dataset consists of 381 samples and 21 features.

### 4.1.2. Explanatory Data Analysis

In this section of the Thesis the descriptive statistics along with visualisation for each feature will be presented to better understand the dataset. The first feature is about the target variable and the commuting mode to work for the respondents. It is evident that there is a balance among private vehicle, walk and bus users with over 100 observations for each. Note that private vehicle includes both commuters via car and motorcycle. Displayed below is the count plot of the feature, accompanied by the corresponding percentages for each mode.

A graph of a number of different colored bars

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| **Mode** | **Percentage** |
| **Bus** | **32.02%** |
| **Walk** | **31.75%** |
| **Private Vehicle** | **36.22%** |

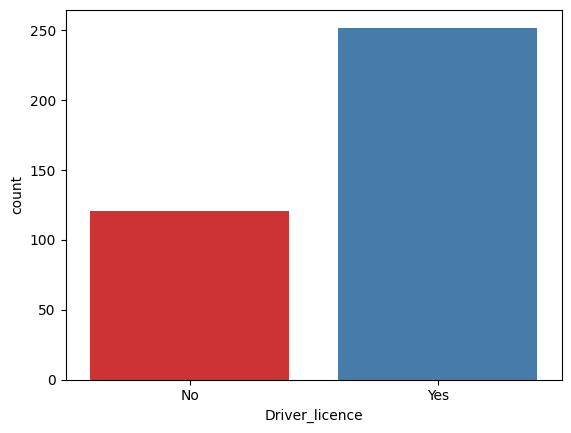
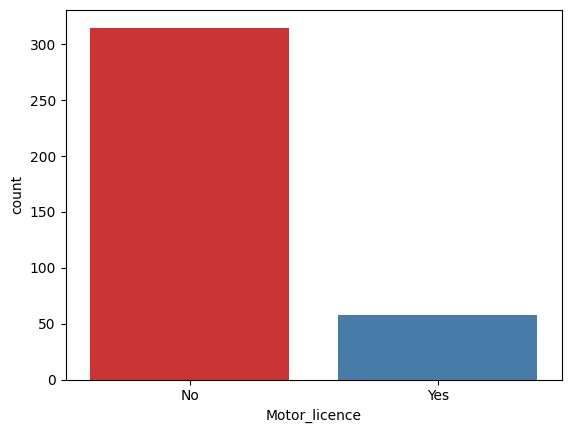
The following feature refers to the gender of the respondents. It is apparent that there is a relatively even distribution among the responses. To be more specific, there are slightly over 175 observations for both men and women, with women having a slim numerical edge (51.18%). The grouped plot also illustrates that a higher proportion of women use the bus compared to men. In contrast, men tend to use private vehicles slightly more than women.

A graph of different colored bars

Description automatically generated

|  |  |
| --- | --- |
| **Gender** | **Percentage** |
| **Women** | **51.18 %** |
| **Men** | **48.82 %** |

The subsequent set of features pertains to whether the respondent possesses driving licences. The options for car and motor licences only allow for "yes" or "no" responses. The plots make it apparent that most respondents possess a car licence compared to motor licence users.



|  |  |  |
| --- | --- | --- |
|  | **Driver licence (car)** | **Motor licence** |
| **Yes** | **68,24 %** | **84.25 %** |
| **No** | **31.75 %** | **15.74 %** |

The following group of features pertains to whether the respondent has access to bike, or skate. Like the preceding set, these features only permit "yes" or "no" as possible answers. In both situation most of the respondents imply not possessing any of those.

A red and blue bar graph

Description automatically generatedA red and blue bar graph

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | **Bike access** | **Skate access** |
| **Yes** | **18.11 %** | **5.73%** |
| **No** | **81.88 %** | **94.22 %** |

|  |  |
| --- | --- |
| **Depart time** | **Percentage** |
| **06.00-09.00** | **46.45 %** |
| **09.00-12.00** | **30.18 %** |
| **12.00-15.00** | **13.39 %** |
| **15.00-18.00** | **6.56 %** |
| **18.00-21.00** | **2.09 %** |
| **00.00-03.00** | **0.78 %** |
| **03.00-06.00** | **0.52 %** |

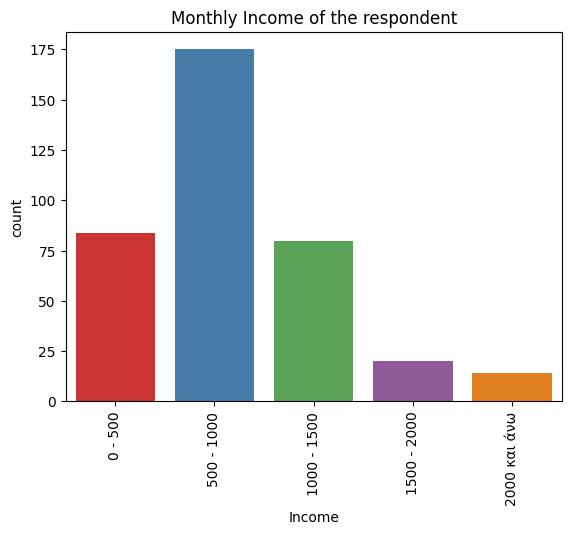
The next feature concerns the departure time of the respondent for their work commute. As depicted in the plot below, it is apparent that most respondents commute to work during the morning peak hours between 06:00 and 12:00. There are also a few instances where respondents commute during the evening hours between 12:00 and 18:00, while the minority proportion commutes to work at off hours during 18:00 - 21.00 and 00.00 - 06.00 past midnight. There are no instances where respondents depart between 21.00 and 00.00.

A graph with different colored bars

Description automatically generated

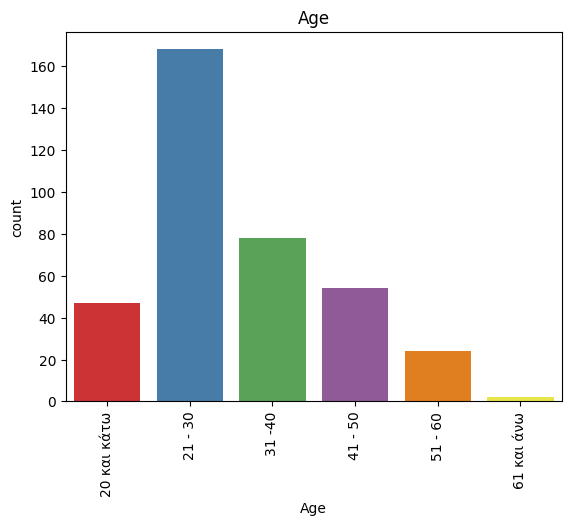
The following question refers to the monthly income of the respondent. As depicted in the graph, most respondents have a monthly income between 500 and 1000£, followed by those with incomes between 0 and 500£. Additionally, there is a substantial proportion of respondents with an income exceeding the 1000£ mark, with a few observations even surpassing 2000£.

|  |  |
| --- | --- |
| **Income** | **Percentage** |
| **0 - 500** | **22.30 %** |
| **500 - 1000** | **46.93 %** |
| **1000 - 1500** | **22.30%** |
| **1500 - 2000** | **5.74 %** |
| **2000 and more** | **3.67 %** |



The next feature pertains to the age group of the respondent. The vast majority fall within the 21 to 30 years old category, followed by those aged between 31 and 40. For the remaining age groups, the observations are relatively more balanced. The sole exception is the age group of 61 and above, which has fewer than 5 observations.

|  |  |
| --- | --- |
| **Age** | **Percentage** |
| **Below 20** | **12.33 %** |
| **21 - 30** | **44.35%** |
| **31 - 40** | **20.99 %** |
| **41 - 50** | **15.23 %** |
| **51 - 60** | **6.56 %** |
| **More than 61** | **0.52%** |

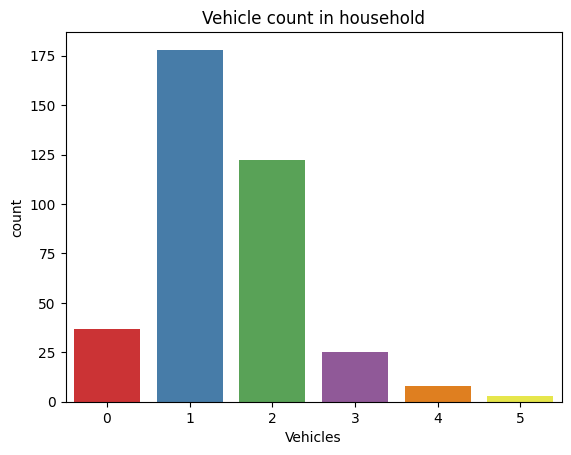


The following feature is about the household size of the respondent. The overwhelming majority indicates residing in households comprising between 2 and 5 members. In addition to this, a minority of respondents suggests living in larger families with 6 or more members, while a small proportion mention being part of a single-member family.

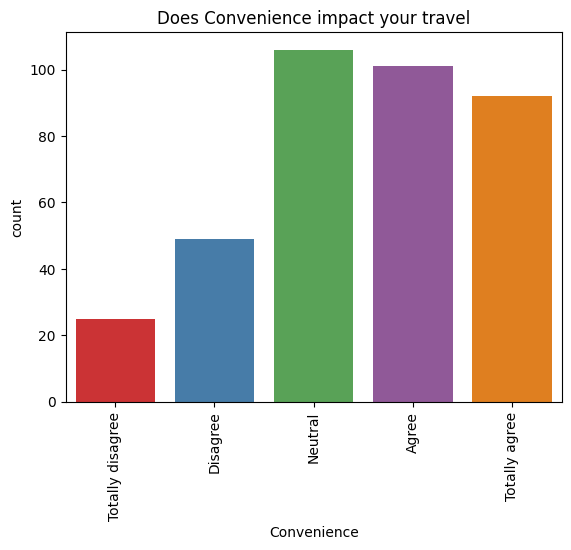


|  |  |
| --- | --- |
| **Household size** | **Percentage** |
| **1** | **3.14 %** |
| **2** | **14.69 %** |
| **3** | **27.82 %** |
| **4** | **35.69 %** |
| **5** | **13.91 %** |
| **6** | **3.14 %** |
| **7** | **0.52 %** |
| **8** | **0.78%** |
| **9** | **0.26 %** |

|  |  |
| --- | --- |
| **Number of vehicles** | **Percentage** |
| **0** | **9.92 %** |
| **1** | **47.72 %** |
| **2** | **32.71 %** |
| **3** | **6.70 %** |
| **4** | **2.14 %** |
| **5** | **0.80 %** |

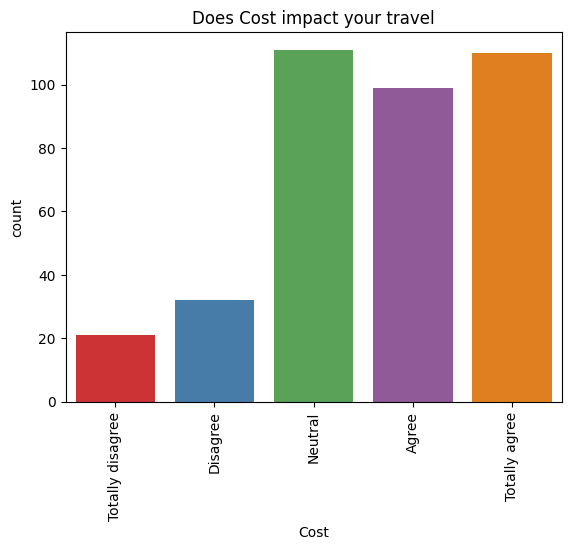
The subsequent feature concerns the number of vehicles in the households of the respondents. Most respondents suggest living in a household with 1 or 2 private vehicles. Additionally, a minority of respondents mentions residing in a house with no vehicles at all, while a few observations suggest households with 4 or 5 vehicles.

The next series of questions addresses the impact of specific factors on the commuting behaviour of the respondent. Starting with convenience as an influencing factor, a significant number of respondents expressed either "agree" or "totally agree" that the convenience of transportation affects their commuting behaviour to work. Additionally, there is a substantial proportion of respondents indicating a "neutral" response to the impact of convenience. Fewer observations are categorised as "disagree" or "totally disagree."



|  |  |
| --- | --- |
| **Impact of Convenience** | **Percentage** |
| **Totally Disagree** | **6.56 %** |
| **Disagree** | **12.86 %** |
| **Neutral** | **28.34 %** |
| **Agree** | **27.29 %** |
| **Totally Agree** | **24.93 %** |

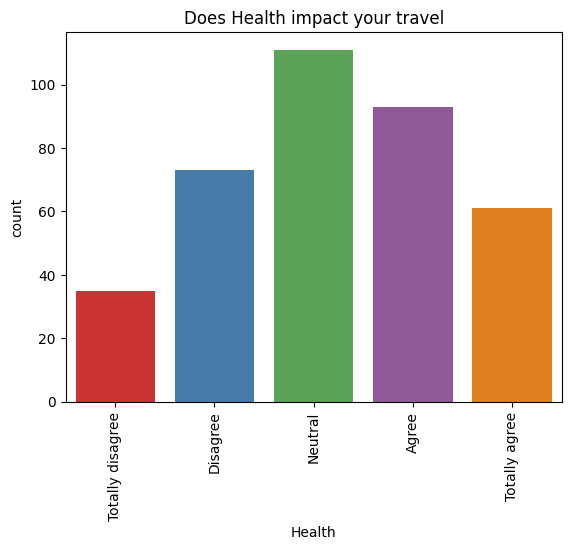
The next feature refers to the impact of transportation cost in commuting behaviour. Most of the respondents also expressed “Agree” and “Totally Agree” that cost is a significant factor influencing their transits to work. Additionally, a significant portion expressed a "neutral" position regarding this aspect. In contrast, only a small number of responses suggested that cost does not impact their commuting behaviour, as reflected in the "Disagree" and "Totally Disagree" categories.



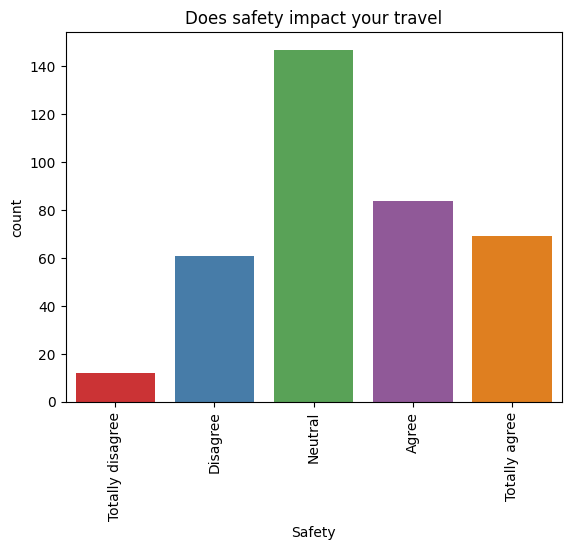
|  |  |
| --- | --- |
| **Impact of Cost** | **Percentage** |
| **Totally Disagree** | **5.51 %** |
| **Disagree** | **8.39 %** |
| **Neutral** | **29.92 %** |
| **Agree** | **25.98 %** |
| **Totally Agree** | **30.18 %** |

The following factor is about physical exercise and health. Specifically, it refers to how physical exercise such as walking and consequently health concerns in general affect the commuter’s transit to work. The results are quite balanced with respondents implying both negative and positive categories. Also, a significant number of respondents also indicated a “neutral” position regarding health impact.

|  |  |
| --- | --- |
| **Impact of Health** | **Percentage** |
| **Totally Disagree** | **9.44 %** |
| **Disagree** | **19.42 %** |
| **Neutral** | **29.92 %** |
| **Agree** | **24.93 %** |
| **Totally Agree** | **16.27 %** |

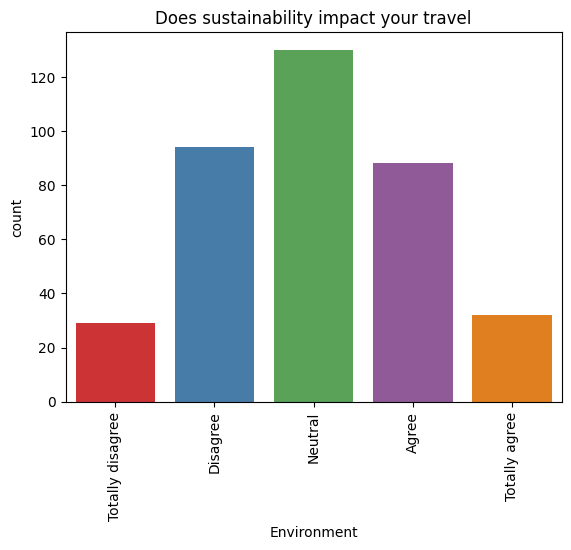


The next feature pertains to the factor of transportation safety. Once again, most respondents express a neutral stance regarding the impact of safety on their commuting behaviour to work. Additionally, a significant proportion reacts positively to the importance of this factor. A smaller number of respondents indicate that safety does not influence their transit behaviour.



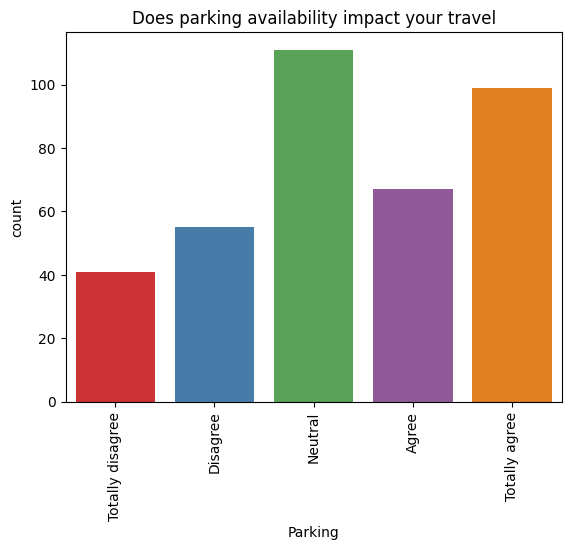
|  |  |
| --- | --- |
| **Impact of Safety** | **Percentage** |
| **Totally Disagree** | **3.41 %** |
| **Disagree** | **16.01 %** |
| **Neutral** | **39.10 %** |
| **Agree** | **22.57 %** |
| **Totally Agree** | **18,89 %** |

The subsequent feature focuses on the influence of environmental concerns. In a manner akin to health and physical exercise, respondents exhibit a balanced stance, with most observations equally distributed across positive and negative categories. Once more, many responses align with a neutral stance for this specific factor.

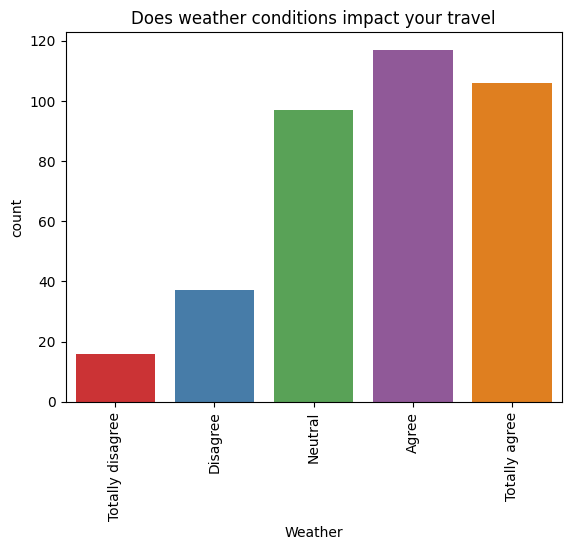


|  |  |
| --- | --- |
| **Impact of Environmental concerns** | **Percentage** |
| **Totally Disagree** | **7.87 %** |
| **Disagree** | **24.93%** |
| **Neutral** | **34.64 %** |
| **Agree** | **23.88 %** |
| **Totally Agree** | **8.66 %** |

Transitioning to the next factor, it concerns the impact of parking availability on commuting behaviour to work. In this context, numerous observations fall into the positive categories, particularly the "totally agree" option, with the "neutral" category being the second most common. The negative categories encompass fewer observations regarding the impact of this factor.

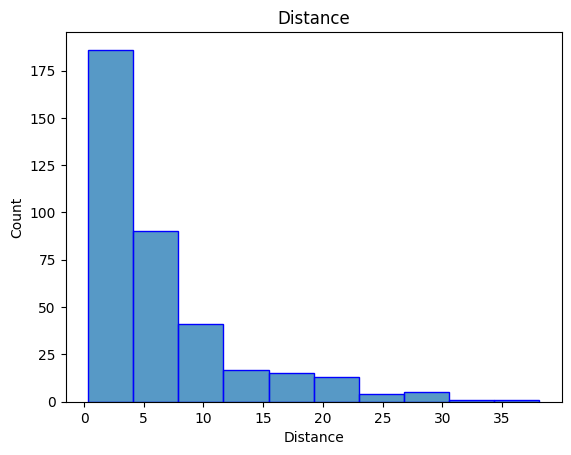
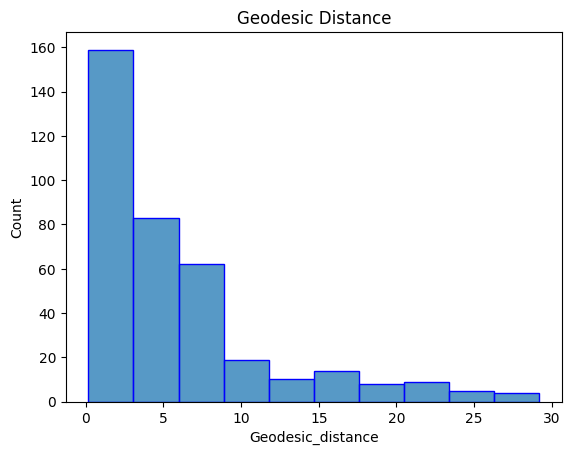


|  |  |
| --- | --- |
| **Impact of parking availability** | **Percentage** |
| **Totally Disagree** | **11.02 %** |
| **Disagree** | **14.69 %** |
| **Neutral** | **29.39 %** |
| **Agree** | **18.37 %** |
| **Totally Agree** | **26.50 %** |

The final factor concerns the impact of weather conditions on commuting to work. In this aspect, there is a notable contrast between positive and negative responses, with most respondents falling into the "agree" and "totally agree" categories. A substantial proportion also indicated a neutral stance on the factor, while both the "disagree" and "totally disagree" categories represent vast minorities.

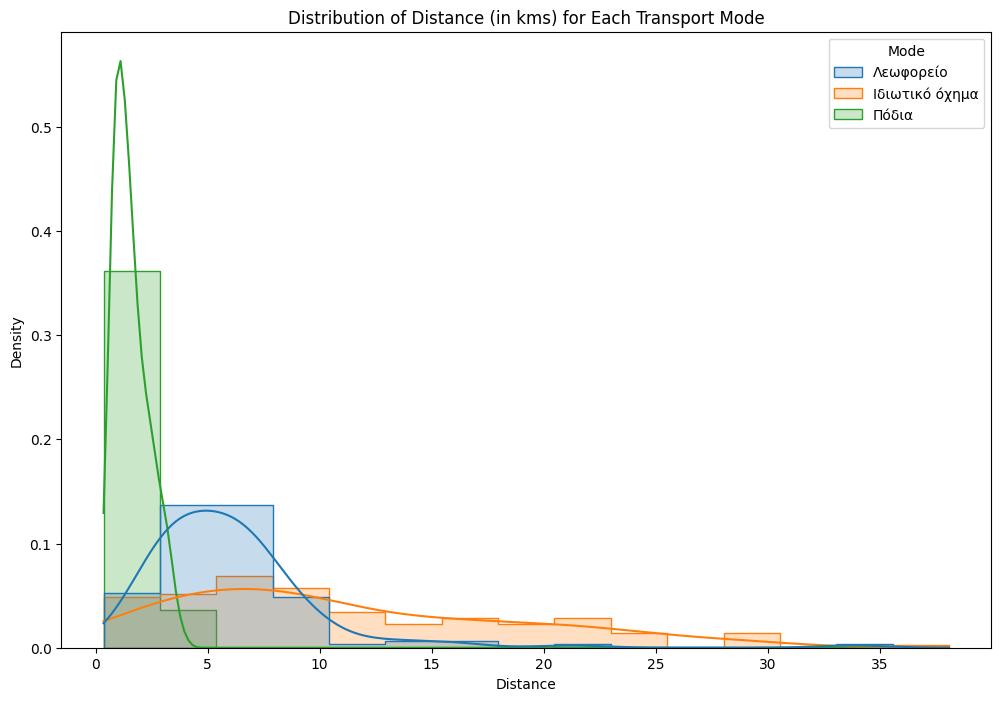
|  |  |
| --- | --- |
| **Impact of weather** | **Percentage** |
| **Totally Disagree** | **4.29 %** |
| **Disagree** | **14.74 %** |
| **Neutral** | **29.76 %** |
| **Agree** | **17.96 %** |
| **Totally Agree** | **26.54 %** |

The subsequent set of features pertains to the distance metrics calculated between the home and work addresses. The "Distance" feature signifies the optimal travel distance manually obtained from Google Maps, whereas the "Geodesic Distance" represents the distance automatically calculated by the Geopy library in Python. Both distances are expressed in kilometres. The histograms for both metrics, along with their descriptive statistics, are depicted below.



|  |  |  |
| --- | --- | --- |
|  | **Distance** | **Geodesic Distance** |
| **Mean** | **6.48** | **5.97** |
| **std** | **6.51** | **6.01** |
| **min** | **0.35** | **0.16** |
| **25 %** | **1.9** | **1.73** |
| **50 %** | **4.3** | **4.04** |
| **75 %** | **8.1** | **7.41** |
| **max** | **38.1** | **29.2** |

The graphs reveal that most observations are concentrated in the initial classes, ranging from 0.1 to 8 kilometres, regardless of the distance metric calculated. Furthermore, both distributions exhibit positive skewness while all the descriptive statistics for geodesic distance appear slightly lower when compared to optimal travel distance. The graphs below show the density of each mode for both distance metrics. Density represents the concentration of data points for a continuous variable in a normalized version over count plots. As illustrated on the plots below the concentration of data for walk is mostly within 0 to 5 kilometres while for bus is within 0 and 10 kilometres. For private vehicle the distribution is more spread across 0 and 35 kilometres.

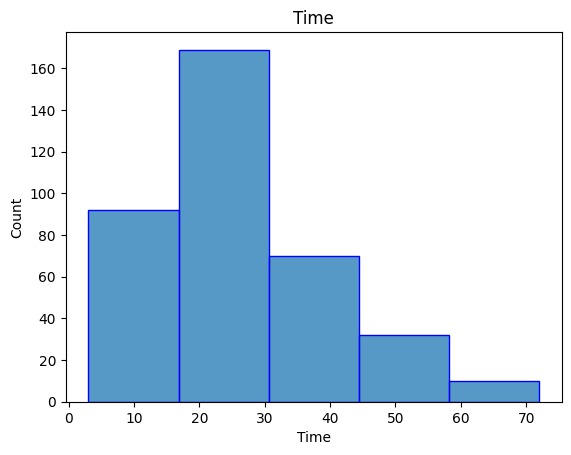


A graph of a distribution of a number of vehicles

Description automatically generated

The final feature concerns the time required (in minutes) for respondents to commute to work, measured in minutes. It can be observed that it follows a distribution like the distance features, indicating a positive skew, with many commutes occurring in the initial classes of approximately 3 to 45 minutes. The histogram, along with the descriptive statistics, are depicted below.

|  |  |
| --- | --- |
|  | **Time** |
| **Mean** | **25.78** |
| **std** | **13.3** |
| **min** | **3** |
| **25 %** | **16** |
| **50 %** | **23** |
| **75 %** | **34** |
| **max** | **72** |



The density plot was again constructed for time and each transport mode. For bus density is more spread across the X-axis while walking and private vehicles are more concentrated within 5- and 40-minutes commute time.

A graph of a number of different colored lines

Description automatically generated

Additional graphs were also generated, to better understand the data. The following graph illustrates the departure time grouped by transport mode. At “peak” hour, typically between 06.00 and 09.00 most of the observations are for private vehicles. Though, as departure time progresses bus and walk commuters overcome those who commute with private vehicles, specifically during 12.00 and 15.00.

A graph of different colored bars

Description automatically generated

The subsequent graph displays the age of the respondents categorized by transport mode. It is apparent that most respondents below 20 years old refrain from commuting to work using a private vehicle. In contrast, as age increases, there is a tendency for individuals to commute to work using a private vehicle. Fewer people opt for bus or walking as their mode of commuting beyond the age of 41.

A graph of different colored bars

Description automatically generated

The following graph illustrates the income range grouped by transport mode. The trend is likewise the age of the respondent. When income falls within the 0 to 500 range, commuters predominantly opt for either bus or walking. Conversely, as income increases, private vehicles become the apparent choice for commuting.

A graph of a number of people

Description automatically generated with medium confidence

### 4.1.3. Data Preprocess

To prepare the data for the classification model, a preprocessing step is necessary to ensure the models function effectively. It is crucial to encode all features into numerical form, as models can only process numerical data. Firstly, low frequencies in some categories were grouped so to not lose information from the dataset. More specifically for Age, Income and Depart time low frequencies were grouped as depicted below.

A screenshot of a computer program

Description automatically generated

The dataset was initially shuffled using the Shuffle() function of sklearn, while a train test split was then implemented of 60:40 ratio. The training set will be used to train the models while the test set will be used to evaluate them. The train set contains 228 samples, while 153 samples belong to the test set. The encoding process was conducted for both sets separately.

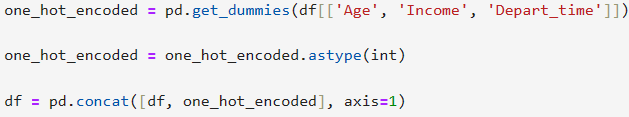
The initial step was to encode the target variable (Mode). The values were encoded into **private** **vehicle - 0, bus - 1 and walk - 2**.

Additionally, the impact factors (cost, convenience, safety, health, environment, parking, weather) were encoded into an ordinal scale of 1-5. More specifically the encoding is illustrated below:



Furthermore, all the binary features were encoded into 0 and 1. This process essentially assigns 0 for "No" and 1 for "Yes" observations. Gender is also part of the label encoder process with 0 assigned to "Males" and 1 to "Females".

The last features that require encoding are Age, Income and Depart\_time. For those features one-hot encoding process was used. This procedure essentially generates new binary features for each potential answer, assigning values of 0 to indicate the absence of that value in the specific row and 1 to imply the presence of that value in the row. The process is illustrated below.



An example of the results from the one hot encoding process is depicted below.

A screenshot of a computer

Description automatically generated

The whole process was repeated two times, one using the Distance feature and a second one using the Geodesic Distance. Essentially two training and test sets were created for evaluating the models using a different distance metric.

The next step was to display the correlation matrix. It is crucial to examine correlations among various features to determine which ones to retain and which ones to drop prior to the modelling application. These decisions rely on the correlation coefficient values, which can range from -1 to 1. Values of -1 and 1 indicate a perfect correlation (positive or negative) between variables, potentially leading to multicollinearity issues that could hinder the performance of the models. Hence, highly correlated features should be eliminated before the application process.

A colorful squares with white and black text

Description automatically generated with medium confidence

From the figure above, it is identified that there are strong correlations in a few cases. Distance and Geodesic\_distance with a value of 0.91, followed by Driver\_licence and Age with 0.44. Only one the Distance metrics will be used for training the models each time. The "vehicles" feature will also be excluded because, when it takes a value of 0 (indicating no household vehicles), it introduces a bias in the classifier, making it easier to classify non-private vehicle observations. Skate, Driver licence and Motor licence features will also be excluded from the models for the same reason. Additionally, to review multicollinearity issues the VIF (Variance Inflator Factor) function was used. VIF measures the severity of multicollinearity, so each feature is assigned a value. Values above 10 raise concerns about the correlation of the features and should be removed. The results of the first attempt using VIF from statsmodels library are depicted below.

A screenshot of a computer

Description automatically generated

The occurrence of "inf" values in certain features points to multicollinearity issues, particularly in one hot encoded features. This arises because, during the one-hot encoding process, one of the resulting binary features must be omitted to prevent multicollinearity. Consequently, Depart time 06:00-00, Income 500-1000, and Age 21-30 were excluded. The VIF function was subsequently reapplied, yielding the following values.

A screenshot of a computer

Description automatically generated

Now, with all VIF values below 5, these finalized features are set to be utilized in the models. Before applying the models, the MinMaxScaler function of sklearn was used to normalize the data.

### 4.1.4. Decision Tree

The first model that was applied was that of the decision tree. The DecisionTreeClassifier(random\_state = 1) function was used to create the model. The random\_state parameter controls the randomness and ensures reproducibility in the results. The model was fitted on the training set using the fit() command and consequently evaluated on the test set using the predict() command. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.86 | 0.93 | 0.89 | 0.84 |
| Bus | 0.83 | 0.69 | 0.76 |
| Walk | 0.83 | 0.90 | 0.86 |
| Macro Average | 0.84 | 0.84 | 0.84 |

A blue squares with white text

Description automatically generated

The overall accuracy of the model is 0.84 while the max\_depth of the tree was found to be 8. The max depth parameter controls the maximum possible length of the path from the root node to a leaf node. The model correctly predicted 44 out of 49 (0.90) instances as “walk” when the true label was “walk”. For classes “private vehicle” and “bus” recall was found 0.93 and 0.69 respectively. The highest precision is for class “private vehicle” (0.86) indicating that of the 59 predictions the model made as “private vehicle” 51 belonged in that class. The biggest misclassification occurs between walk - bus, where the model predicted 8 instances as “walk” while the true label was “bus”. Although the model's performance appears mediocre, the training accuracy is 1 (100%), suggesting an overfitting issue. To address this concern, a 10-fold cross-validation was implemented to assess whether tree pruning, which involves reducing the size of the tree, enhances the model's performance. The results of cross validation are presented below.

A graph with a line going up

Description automatically generated

From the validation results it is identified that accuracy is quite similar for depth equal to 6 and 7. The tree was pruned at max depth value of 6 and was revaluated on the test set. The results are illustrated on the classification report below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.86 | 0.91 | 0.88 | 0.86 |
| Bus | 0.86 | 0.73 | 0.79 |
| Walk | 0.87 | 0.94 | 0.90 |
| Macro Average | 0.86 | 0.86 | 0.86 |

Compared to the original tree model, the overall accuracy has been increased from 0.84 to 0.86. Recalls for both bus and walk increased. More particularly, for bus recall increased from 0.69 to 0.73, while for walk it increased from 0.90 to 0.94. In contrast, recall for private vehicle decreased from 0.93 to 0.91. Overall reducing the max depth parameter slightly increased the performance, though training accuracy remains high at 0.98. Hence, additional parameters were tuned for the model. More specifically, min\_samples\_split and min\_samples\_leaf parameters were also tuned via GridSearchCV() function. Min\_samples\_split refers to the number of samples required to split an internal node. In contrast, min\_samples\_split refers to the number of samples required to be in a leaf mode. The set of parameters for the validation process are depicted below:



The validation process wielded the following results:

**max\_depth = 5**, **min\_samples\_split = 2**, **min\_samples\_leaf = 5**. The best model was then retrieved and evaluated on the test set. The results are depicted below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.91 | 0.87 | 0.89 | 0.87 |
| Bus | 0.86 | 0.76 | 0.80 |
| Walk | 0.84 | 0.98 | 0.91 |
| Macro Average | 0.87 | 0.87 | 0.87 |

A screenshot of a graph

Description automatically generated

The overall accuracy of the decision tree increased from 0.86 to 0.87. Simultaneously, the training accuracy decreased from 0.98 to 0.92. Compared to the previous tree, recall for "bus" increased from 0.73 to 0.76, while for walk increased from 0.94 to 0.98. Furthermore, the precision for "private vehicle" also increased from 0.86 to 0.91. The most significant misclassification still occurred for walk and bus labels, where the model predicted 7 instances as "walk" but the true label was "bus". Below the tree structure of the pruned tree is illustrated.

A diagram of a company structure

Description automatically generated

Finally, the feature importances of the model were retrieved using the feature\_importances\_ function. Results are illustrated below:

A graph with a bar graph

Description automatically generated

The feature importance essentially shows how much influence each feature has on prediction and more specifically, how much each feature contributed to the splits that the tree made. The plot above reveals that Distance and Time features are identified as the most crucial factors for the models, followed by Convenience and Health. It is noteworthy that only five features out of the 23 available were selected to construct the decision tree.

The next experimentation was using the Geodesic\_distance instead of the Distance feature. The model was trained on the training data and evaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.83 | 0.78 | 0.80 | 0.79 |
| Bus | 0.78 | 0.71 | 0.74 |
| Walk | 0.77 | 0.88 | 0.82 |
| Macro Average | 0.79 | 0.79 | 0.79 |

A screenshot of a graph

Description automatically generated

The accuracy of the model using the geodesic\_distance dropped at 0.79. The performance for each individual class has also drastically dropped compared to the previous model. The biggest misclassification occurs between “bus” and “private vehicle” and “walk” – “bus” with 7 missclassified cases each. Overall swapping the distance metrics reduced the performance of the decision tree, while the training accuracy is high at 1 indicating an overfitting issue. The tree was again pruned using cross validation. The results of the tree depth are illustrated below.

A graph with a line going up

Description automatically generated

Since a value 4 for tree depth yields the best results, the model was pruned and was reevaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.87 | 0.87 | 0.87 | 0.82 |
| Bus | 0.73 | 0.78 | 0.75 |
| Walk | 0.87 | 0.82 | 0.84 |
| Macro Average | 0.82 | 0.82 | 0.82 |

The overall accuracy of the model increased from 0.79 to 0.82 while the training accuracy dropped from 1 to 0.89. Recalls for both private vehicle and bus increased from 0.78 to 0.87 and 0.71 to 0.78 respectively. In contrast, recall for walk dropped from 0.88 to 0.82 with a trade off in precision which increased from 0.77 to 0.87. While reducing the max depth of the tree slightly improved the performance, additional parameters were tuned like the previous models. Again, a grid search was run using the min\_samples\_leaf and min\_samples\_split parameters. The parameter values are illustrated below.

A screenshot of a computer code

Description automatically generated

The best new parameters were found for max\_depth = 5, min\_samples\_leaf=6 and min\_samples\_split=2. The new tuned model was then evaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.90 | 0.85 | 0.88 | 0.84 |
| Bus | 0.77 | 0.82 | 0.79 |
| Walk | 0.84 | 0.84 | 0.84 |
| Macro Average | 0.84 | 0.84 | 0.84 |

A screenshot of a graph

Description automatically generated

Recall for bus mode increased from 0.78 to 0.82, while precision also increased from 0.73 to 0.77. Recall also increased for walk mode from 0.82 to 0.84. In contrast, recall for private vehicle dropped from 0.87 to 0.87. The overall accuracy of the model increased from 0.82 to 0.84, while the training accuracy dropped from 0.89 to 0.87. The tree structure for the pruned tree using the geodesic distance is illustrated below.

A diagram of a company structure

Description automatically generated

Finally, the feature importances were extracted for the tree using the geodesic distance. The results are depicted below.

A graph with blue bars

Description automatically generated

Following the trend of the previous model, geodesic distance and time are the most important features for constructing the tree, followed by health. The model used 7 features in total, compared to 5 features used in the previous pruned model that utilized the Distance metric.

The following table illustrates the metrics for the two models using different distance metrics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Pruned Tree (Distance) | | | Pruned Tree (Geodesic distance) | | |
|  | Private vehicle | Bus | Walk | Private vehicle | Bus | Walk |
| Precision | 0.91 | 0.86 | 0.84 | 0.90 | 0.77 | 0.84 |
| Recall | 0.87 | 0.76 | 0.98 | 0.85 | 0.82 | 0.84 |
| F1-score | 0.89 | 0.80 | 0.91 | 0.88 | 0.79 | 0.84 |
| Accuracy | 0.87 | | | 0.84 | | |

The first model (using distance) demonstrates better performance for classes private vehicle and bus, with increased metrics in precision, recall, and f1. The overall accuracy is also higher with 0,87 compared to 0.84. However, utilizing Geodesic distance demonstrates better recall for bus, with 0,82 compared to 0.76. The first model demonstrates a more balanced performance overall.

### 4.1.5. Random Forest

The next model that was implemented was that of Random Forest. The RandomForestClassifier() function was used to create the model. The model was fitted on the training data, followed by evaluation on the test set. The results are illustrated below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.96 | 0.93 | 0.94 | 0.92 |
| Bus | 0.88 | 0.90 | 0.89 |
| Walk | 0.92 | 0.94 | 0.93 |
| Macro Average | 0.92 | 0.92 | 0.92 |

A screenshot of a graph

Description automatically generated

The overall accuracy of the basic model is 0.92, correctly predicting 141 out of 153 total instances. Additionally, the model accurately predicted 44 out of 49 (0.90) cases as "bus" when the true label was "bus." Recall for private vehicle and walk was 0.93 and 0.94 respectively. The highest precision is for private vehicle mode accurately predicting 51 out of 53 predictions made as private vehicle. The default values of the model are max\_features = "auto" and n\_estimators = 100 and max\_depth=’None’. The max\_features parameter refers to the number of features considered each time a split occurs on the decision tree, while “auto” is the square root of the total number of features. In contrast, the n\_estimators refer to the number of trees the model builds. Max\_depth, which is the depth of each individual tree, is set to None indicating that each individual tree grows without limit. Similar to the decision tree, the GridSearchCV() function was used to tune the model and find the optimal parameters attempting to increase the performance of the model. The process was run for each parameter individually to identify optimal area for a final grid search. Since the train set contains 22 total features, max features parameter was set at a range between 2 and 23. The validation results are depicted below.

A graph with blue lines and dots

Description automatically generated

Looking at the plot above, it is evident that the validation accuracy varies when the number of features is above 5, with the best overall performance to be for 14 number of features. The next parameter tuned was the number of estimators. The range was set between 50 and 300 number of trees. The validation results are illustrated below.

A graph showing a number of trees

Description automatically generated

The optimal area, where the validation accuracy is higher, is between 130 and 175 number of trees. The final parameter that was tuned was the max depth of the individual tree. The results are illustrated below.

A graph with a line

Description automatically generated

The random forest model performs best for a tree depth 7. After finding the optimal areas, a final grid search was executed with the following parameter values.

'n\_estimators': list(range(150, 176)),

'max\_features': [10, 11, 12, 13, 14],

'max\_depth': [7]}

The results of the final grid search were **max\_features = 14** and **n\_estimators = 150, and max\_depth=7**. The validated model was then retrieved and evaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.98 | 0.96 | 0.97 | 0.94 |
| Bus | 0.92 | 0.90 | 0.91 |
| Walk | 0.92 | 0.96 | 0.94 |
| Macro Average | 0.94 | 0.94 | 0.94 |

A screenshot of a graph

Description automatically generated

The overall accuracy of the tuned model increased from 0.92 to 0.94. Recall and precision for private vehicle increased from 0.96 and 0.93 to 0.98 and 0.96 respectively. For bus mode, recall remained the same at 0.90 while precision increased from 0.88 to 0.92. For walk mode recall also increased from 0.94 to 0.96 while precision remained at 0.92. The tuned model illustrates a better performance overall compared to the default model. Similarly with the decision tree, the feature importance was retrieved for the tuned model. The results are illustrated below.

A graph with blue bars

Description automatically generated

The model prioritises Distance, Time, Health, Convenience, Cost, Safety and Weather as the most crucial features. Unlike the simple tree model, the random forest has essentially utilised all parameters for split criteria.

The last experimentation was swapping Distance with the Geodesic Distance feature. A new model was created, fitted on the training data, and consequently evaluated on the test set. The results are depicted below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.92 | 0.89 | 0.91 | 0.90 |
| Bus | 0.86 | 0.90 | 0.88 |
| Walk | 0.90 | 0.90 | 0.90 |
| Macro Average | 0.90 | 0.90 | 0.90 |

A blue squares with white text

Description automatically generated

The overall accuracy of the model was 0.90, correctly predicting 137 out of 153 instances. Recall for private vehicle was 0.89, retrieving 49 out of 55 relevant instances. Recall for bus and walk modes was 0.90 retrieving 44 out of 49 relevant instances respectively. The highest precision is for private vehicle with 0.92. The next step was to tune the parameters of the model attempting to increase the performance. Again, max\_depth, number of estimators and max features were configured via grid search. The results are illustrated below.

A graph with blue lines

Description automatically generated

A graph showing a number of trees

Description automatically generated

A graph with a line

Description automatically generated

Afte finding the optimal areas, a final grid search was configured using the following parameter values.

'n\_estimators': list(range(75, 91)),

'max\_features': [8, 11, 13],

'max\_depth': [7]}

The best new parameters were found for max\_depth = 7, n\_estimators = 83 and max\_features = 11. The tuned model was then evaluated on the test set. The results are depicted below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.93 | 0.95 | 0.94 | 0.91 |
| Bus | 0.93 | 0.86 | 0.89 |
| Walk | 0.87 | 0.92 | 0.89 |
| Macro Average | 0.91 | 0.91 | 0.91 |

A blue squares with white text

Description automatically generated

The overall accuracy of the tuned model increased from 0.90 to 0.91. Recall score increased for private vehicle from 0.89 to 0.95, while for walk increased from 0.90 to 0.92. Precision also increased for both private vehicle and bus at 0.93. In contrast, recall for bus mode decreased from 0.90 to 0.86. the highest misclassification occurs between walk and bus where the model predicted 5 instances as walk while the true label was bus. The feature importances for the model construction were also extracted. The results are illustrated in the plot below. The trend is likewise the previous Random Forest model, as both distances followed by time and Health, Cost and Convenience are the most important features.

A graph with blue bars

Description automatically generated

The table below shows the two models utilizing distance and geodesic distance metrics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Random Forest (Distance) | | | Random Forest (Geodesic distance) | | |
|  | Private vehicle | Bus | Walk | Private vehicle | Bus | Walk |
| Precision | 0.98 | 0.92 | 0.92 | 0.93 | 0.93 | 0.87 |
| Recall | 0.96 | 0.90 | 0.96 | 0.95 | 0.86 | 0.92 |
| F1-score | 0.97 | 0.91 | 0.94 | 0.94 | 0.92 | 0.89 |
| Accuracy | 0.94 | | | 0.91 | | |

The Random Forest model, utilizing the Distance metric, performs better overall compared to the model using Geodesic Distance.

### 4.1.6. XGBoost

The next model applied was that of XGboost. The model was created using the **xgb.XGBClassifier(objective='multi:softmax', num\_class=3, random\_state=42)** function.

The objective=”multi:softmax” parameter indicates that the task of the model is to classify more than two classes, while the num\_class = 3 parameter indicates the number of classes. Similarly to random forest, the random\_state parameter is set to achieve reproducibility of the results. The model was then fitted on the training set and consequently evaluated on the test set. The results are depicted below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.95 | 0.98 | 0.96 | 0.94 |
| Bus | 0.95 | 0.86 | 0.90 |
| Walk | 0.92 | 0.98 | 0.95 |
| Macro Average | 0.94 | 0.94 | 0.94 |

A screenshot of a graph

Description automatically generated

The overall accuracy of the model was 0.94, correctly predicting 144 out of 153 total instances. Recall for both “private vehicle” and “walk” was found 0.98 respectively, while precision was highest for “private vehicle” and “bus” at 0.95. In contrast, recall for bus in the lowest of 0.86, retrieving 42 out of 49 relevant instances. The biggest misclassification occurs for “bus” and “walk”, as the model predicted 4 instances as “walk” while the true label was “bus”. Although the model's performance appears satisfactory, various sets of parameters were modified to assess if performance could be enhanced. The default values of the model are the following: **n\_estimators = 100, subsample = 1, colsample\_bytree =1 , learning\_rate = 0.3**. The number of estimators refers to the number of trees built like Random Forest. The subsample parameter refers to the fraction of the training data that are sampled during the training phase. The colsample\_bytree parameter refers to the fraction of features that are sampled for each tree. Finally, the learning\_rate parameter refers to the contribution of each tree to the overall model, while affecting how quickly or slowly the model adapts to the patterns in the training data. Three additional parameters were also explored, gamma, lambda and max tree depth. Gamma influences the tree complexity controlling a trade-off between complexity and overfitting. Lambda is the L2 regularization term penalizing large tree weights. The max\_deth parameter like random forest and decision tree controls the individual tree size. Those parameters were configured through the GridSearchCV() function individually to create a final grid and evaluate the model.

Initially, a grid search was conducted to test the number of estimators within the range of 50 to 300 trees. The outcomes are illustrated below.

A graph with blue lines

Description automatically generated

The overall validation accuracy is higher between 200 and 250 number of trees. The next parameter that was tested was that of learning\_rate with values from 0.1 up to 1. The results are illustrated below.

A graph with a line

Description automatically generated

The results indicate best validation accuracy for learning\_rate = 0.4. The next parameter tested was subsamples, with values from 0.1 to 1. The results are illustrated below.

A graph with a line

Description automatically generated

It is identified that the validation accuracy is best for 0.6. The next parameter tuned was that of colsample\_bytree. The results are illustrated below.

A graph with a line

Description automatically generated

The validation accuracy is best for colsample value of 0.7 and 0.9. The next parameter explored was Gamma with values ranging from 0 to 0.9.

A graph with blue lines

Description automatically generated

The best validation accuracy is for 0.1. The next parameter explored was lambda with values ranging from 0 to 9.

A graph with blue lines

Description automatically generated

The best accuracy is achieved by Lambda value of 8. The last parameter tuned was that of max\_depth. The results are depicted below.

A graph with a line

Description automatically generated

After finding the optimal values a final grid was run to tune the model. While the best accuracy was found for trees of 250 to 300, tha range was set ot 200 to 225 trees. Additionally only the max\_depth parameter was chosen among regularization terms (max\_depth, lambda and gamma). The parameter values that were tuned through grid search are illustrated below.

'n\_estimators' : list(range(200, 226)),

'gamma' : [0],

'reg\_lambda' : [0],

'subsample': [0.8],

'colsample\_bytree' : [0.9],

'learning\_rate': [0.4],

'max\_depth':[2]}

The optimal parameters values from grid search are the following: **n\_estimators = 200, subsample = 0.8, colsample\_bytree = 0.9, learning\_rate = 0.4, max\_depth=2**. The tuned model was then retrieved and evaluated on the test set. The results are depicted below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.96 | 0.96 | 0.96 | 0.94 |
| Bus | 0.94 | 0.90 | 0.92 |
| Walk | 0.92 | 0.96 | 0.94 |
| Macro Average | 0.94 | 0.94 | 0.94 |

A screenshot of a graph

Description automatically generated

The overall accuracy of the tuned model remained the same at 0.94. There is a trade-off for Recall between classes. While for private vehicle and walk it decreased from 0.98 to 0.96, for bus it increased from 0.86 to 0.90. Precision for private vehicle increased from 0.94 to 0.95, while for walk remained the same at 0.92. The feature importance was retrieved for the model. The results are depicted below.

A graph of a bar graph

Description automatically generated with medium confidence

Compared to the previous tree-based models, XGBoost has utilized most of the features, even the binary ones. The most important was whether the Income was more than 1500 or not, followed by Convenience factor. Next was whether the income was between 0 and 500 or not, while Distance which was the most important factor for the previous models, is now ranked 4th. Time, which was also second for both previous models, is now ranked at 10th.

The last experimentation was swapping Distance with Geodesic\_Distance. A new model was created, trained on the train set and consequently evaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.89 | 0.91 | 0.90 | 0.88 |
| Bus | 0.89 | 0.86 | 0.88 |
| Walk | 0.86 | 0.88 | 0.87 |
| Macro Average | 0.88 | 0.88 | 0.88 |

A blue squares with white text

Description automatically generated

The overall accuracy of the model dropped from 0.94 to 0.88 swapping distances. Recall for private vehicle was 0.91, followed by walk at 0.88 and bus with 0.86. Precision is higher for both private vehicle and bus with 0.89. While the overall metrics of the model dropped swapping distances, the parameters were re-tuned to increase the performance of the model. The same parameters as before were tuned to find optimal areas. The validation results are illustrated below.

A graph with blue lines

Description automatically generated

A graph with a line

Description automatically generated

A graph with a line

Description automatically generated

A graph with a line

Description automatically generated

A graph with a line

Description automatically generated

A graph with a line

Description automatically generated

A graph with blue lines and numbers

Description automatically generated

After finding optimal areas, a final grid search was run with the following parameters.

'n\_estimators' : list(range(50, 91)),

'gamma' : [0],

'max\_depth' : [2],

'reg\_lambda' : [6],

'subsample': [0.7],

'colsample\_bytree' : [0.6],

'learning\_rate': [0.2]}

The best parameters were found for colsample = 0.6, learning\_rate=0.2, max\_depth = 2, n\_estimators=75, lambda=6 and subsample = 0.7. The model was then evaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.91 | 0.91 | 0.91 | 0.90 |
| Bus | 0.90 | 0.90 | 0.90 |
| Walk | 0.88 | 0.88 | 0.88 |
| Macro Average | 0.89 | 0.89 | 0.89 |

A screenshot of a graph

Description automatically generated

A graph of a bar graph

Description automatically generatedFor the tuned model, recall for both bus and walk increased at 0.90 and 0.88 respectively. Precision for all the individual classes slightly increased as well. Furthermore, the overall accuracy increased from 0.88 to 0.90 correctly predicting 137 out of 153 total instances. The feature importance was also retrieved for the tuned model. The results are depicted below.

Geodesic distance ranks 1st as the most important feature followed by health, convenience, and weather. Interestingly, while Income 1500 or more was the most important feature for the previous model (utilizing distance metric), now it is ranked 13th. Again, all features contributed for constructing the individual trees. The table below depicts the metrics for both models (Distance vs Geodesic).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | XGBoost (Distance) | | | XGBoost (Geodesic distance) | | |
|  | Private vehicle | Bus | Walk | Private vehicle | Bus | Walk |
| Precision | 0.96 | 0.94 | 0.92 | 0.91 | 0.90 | 0.88 |
| Recall | 0.96 | 0.90 | 0.96 | 0.91 | 0.90 | 0.88 |
| F1-score | 0.96 | 0.92 | 0.94 | 0.91 | 0.90 | 0.88 |
| Accuracy | 0.94 | | | 0.90 | | |

The XGBoost model, utilizing the Distance metric, performs better overall compared to the model using Geodesic Distance.

### 4.1.7. Stacked Model

The last model applied was a Stacked Model. Stacking occurs when multiple models are combined to create an enhanced classifier. For the stacked model to be constructed, base classifiers and a “meta” learner need to be defined. The tree-based models (Decision Tree, Random Forest and XGBoost) were used as the base estimators while Logistic Regression was used as the final estimator. The StackingCVClassifier of mlxtend library was used to create the model. As a first experimentation the default values for all estimators were used. The configuration of the model is illustrated below.

base\_classifier1 **=** RandomForestClassifier(random\_state**=**1,n\_jobs**=-**1)

base\_classifier2 **=** xgb**.**XGBClassifier(objective**=**'multi:softmax', num\_class**=**3, random\_state**=**42)

base\_classifier3 **=** DecisionTreeClassifier(random\_state**=**12)

meta\_classifier **=** LogisticRegression(multi\_class**=**'ovr')

stacking\_classifier **=** StackingCVClassifier(

classifiers**=**[base\_classifier1, base\_classifier2, base\_classifier3],

meta\_classifier**=**meta\_classifier,

cv**=**10,

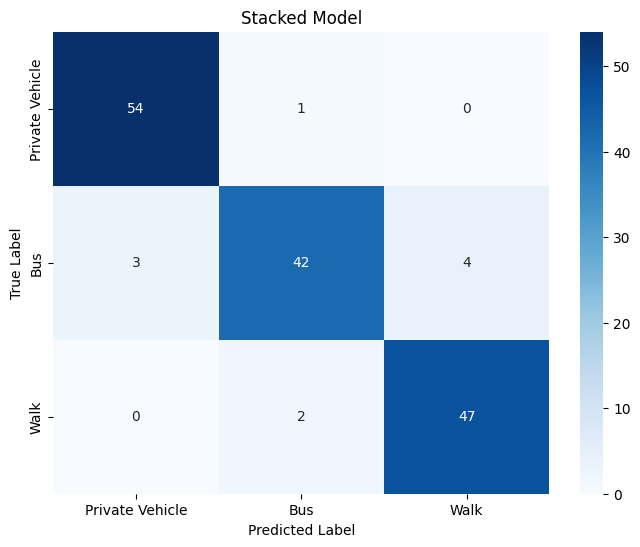
stratify**=True**,

random\_state**=**10

)

The model was initially fitted on the training set and was evaluated on the test set afterwards. The results from the test set are illustrated on the classification report and the confusion matrix below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.95 | 0.98 | 0.96 | 0.93 |
| Bus | 0.93 | 0.86 | 0.89 |
| Walk | 0.92 | 0.96 | 0.94 |
| Macro Average | 0.93 | 0.93 | 0.93 |



The overall accuracy of the model was 0.93. For "Private Vehicle," the model achieves high precision (0.95) and a recall of 0.98. In the "Bus" category, the precision is at 0.93, while recall is at 0.86. For walk, precision and recall are 0.92 and 0.96 respectively. The biggest misclassification occurs between bus and walk, where the model predicted 4 instances as walk while the true label was bus.

The second experimentation was to use the tuned models from the previous chapters. The configuration of the new stacked model is illustrated below.

base\_classifier1 **=** RandomForestClassifier(random\_state**=**1,n\_jobs**=-**1, max\_depth**=**7, max\_features**=**14, n\_estimators**=**150)

base\_classifier2 **=** xgb**.**XGBClassifier(objective**=**'multi:softmax', num\_class**=**3, random\_state**=**42,

colsample\_bytree**=** 0.9, learning\_rate**=** 0.4, max\_depth**=** 2, n\_estimators**=** 200, subsample**=** 0.8)

base\_classifier3 **=** DecisionTreeClassifier(random\_state**=**12, max\_depth**=**5, min\_samples\_leaf **=** 5, min\_samples\_split**=** 2)

meta\_classifier **=** LogisticRegression(multi\_class**=**'ovr')

stacking\_classifier **=** StackingCVClassifier(

classifiers**=**[base\_classifier1, base\_classifier2, base\_classifier3],

meta\_classifier**=**meta\_classifier,

cv**=**10,

stratify**=True**,

random\_state**=**10

)

The model was initially fitted on the training set and was evaluated on the test set afterwards. The results from the test set are illustrated on the classification report and the confusion matrix below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.96 | 0.96 | 0.96 | 0.93 |
| Bus | 0.90 | 0.88 | 0.89 |
| Walk | 0.92 | 0.94 | 0.93 |
| Macro Average | 0.93 | 0.93 | 0.93 |

A blue squares with white text

Description automatically generated

The overall accuracy of the new stacked remained the same at 0.93. There is a trade off in recalls for the classes. For bus it increased from 0.86 to 0.88, while for private vehicle and walk it dropped at 0.96 and 0.94 respectively. Precision for private vehicle also slightly increased from 0.95 to 0.96, while for bus it dropped from 0.93 to 0.90. The biggest misclassification occurs between walk and bus, where the model predicted 4 instances as walk when the true label was bus.

As a second experimentation, again geodesic distance was swapped with Distance and a new stacked model was constructed using the default parameters of the base estimators. The configuration for the model is illustrated below.

base\_classifier1 **=** RandomForestClassifier(random\_state**=**42,n\_jobs**=-**1)

base\_classifier2 **=** xgb**.**XGBClassifier(objective**=**'multi:softmax', num\_class**=**3, random\_state**=**42)

base\_classifier3 **=** DecisionTreeClassifier(random\_state**=**42)

meta\_classifier **=** LogisticRegression(multi\_class**=**'ovr')

stacking\_classifier **=** StackingCVClassifier(

classifiers**=**[base\_classifier1, base\_classifier2, base\_classifier3],

meta\_classifier**=**meta\_classifier,

cv**=**10,

random\_state**=**1

)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.91 | 0.95 | 0.93 | 0.87 |
| Bus | 0.87 | 0.80 | 0.83 |
| Walk | 0.82 | 0.86 | 0.84 |
| Macro Average | 0.87 | 0.87 | 0.87 |

A blue squares with white text

Description automatically generated

The overall accuracy of the model was 0.87. For "Private Vehicle," the model achieves high precision (0.91) and a recall of 0.95. In the "Bus" category, the precision is at 0.87, while recall is lower at 0.80. For walk, precision and recall are also lower at 0.82 and 0.86 respectively. The highest misscassification occurs between walk and bus classes, The overall performance of the model significantly dropped by swapping distances. Again, the tuned models (with Geodesic distance) were used to retrain the model in attempting to improve the performance. The new configuration is illustrated below.

base\_classifier1 **=** RandomForestClassifier(random\_state**=**42,n\_jobs**=-**1, max\_depth**=**7, max\_features**=**11, n\_estimators**=** 99)

base\_classifier2 **=** xgb**.**XGBClassifier(objective**=**'multi:softmax', num\_class**=**3, random\_state**=**42,

colsample\_bytree**=**0.6, learning\_rate**=** 0.2, max\_depth**=** 2, n\_estimators**=**75, reg\_lambda**=** 6, subsample**=**0.7)

base\_classifier3 **=** DecisionTreeClassifier(random\_state**=**42, max\_depth**=** 5)

meta\_classifier **=** LogisticRegression(multi\_class**=**'ovr')

stacking\_classifier **=** StackingCVClassifier(

classifiers**=**[base\_classifier1, base\_classifier2, base\_classifier3],

meta\_classifier**=**meta\_classifier,

cv**=**10,

random\_state**=**1

)

The model was then evaluated on the test set. The results are illustrated below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | Accuracy |
| Private Vehicle | 0.91 | 0.95 | 0.93 | 0.88 |
| Bus | 0.93 | 0.80 | 0.86 |
| Walk | 0.81 | 0.90 | 0.85 |
| Macro Average | 0.89 | 0.88 | 0.88 |

A screenshot of a graph

Description automatically generated

The overall accuracy of the model increased from 0.87 to 0.88. Recall for walk increased from 0.86 to 0.90. Precision also increased for bus from 0.87 to 0.93. For private vehicle all of Precision, Recall and F1 remained the same at 0.91, 0.95 and 0.93 respectively. The highest misclassification still occurs between walk and bus, as the model predicted 8 instances as walk when the true label was bus. The only improvement of the tuned stacked model only occurs for walk label, still falling behind the model utilizing Distance feature.

The table below depicted the metrics of the models utilizing different distance metrics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stacked (Distance) | | | Stacked (Geodesic distance) | | |
|  | Private vehicle | Bus | Walk | Private vehicle | Bus | Walk |
| Precision | 0.96 | 0.90 | 0.92 | 0.91 | 0.93 | 0.93 |
| Recall | 0.96 | 0.88 | 0.94 | 0.95 | 0.80 | 0.86 |
| F1-score | 0.96 | 0.94 | 0.93 | 0.93 | 0.90 | 0.85 |
| Accuracy | 0.93 | | | 0.88 | | |

### 4.1.8. Model Selection

Before assessing the model performance, the ROC curves for the best models were constructed. The ROC curve is a graphical representation that evaluates the trade-off between true positive rate and false positive rate. The curve is generated by calculating true positive rate and false positive rate values across various probability thresholds. Although these curves are typically created for binary classification scenarios, in a multi-classification task, curves are generated for each individual class using a "One vs Rest" approach. For example, in the case of the "private vehicle" class, the true positive rate signifies the ratio of correctly predicted observations belonging to that class. Conversely, the false positive rate for "private vehicle" indicates the ratio of observations falsely predicted as belonging to that class when they actually belong to any of the other classes (bus or walk). The metric representing the model's performance is the area under the curve (AUC), calculated for each class individually in a multiclass task. The AUC macro score is then derived by averaging the AUC scores for each class. This score ranges from 0 to 1, providing an overall assessment of the model's performance. The ROC curves of the models utilizing Distance feature are presented below.

A graph of a curve

Description automatically generatedA graph of a curve

Description automatically generated

A graph of a graph

Description automatically generated

A graph of a graph

Description automatically generated

After applying the 4 models to the training set and evaluating their predicting performance on the test set, while also constructing their ROC curves, a concrete decision must be made on which model performs the best and can be trusted for prediction on new unseen data. For that purpose, a comparative table was constructed highlighting the averages of Precision, Recall, F-score, AUC between classes and the overall accuracy of the best models that were constructed. The models below all use the Distance feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models with Distance | **Decision Tree** | **Random Forest** | **XGBoost** | **Stacked Model** |
| **Precision (Average)** | 0.87 | 0.94 | **0.94** | 0.93 |
| **Recall (Average)** | 0.87 | 0.94 | **0.94** | 0.93 |
| **F-Score (Average)** | 0.87 | 0.94 | **0.94** | 0.93 |
| **AUC (Average)** | 0.924 | 0.984 | **0.986** | 0.96 |
| **Accuracy** | 0.87 | 0.94 | **0.94** | 0.93 |

From the table above it is evident that XGBoost and Random Forest demonstrate the best overall performance for Precision, Recall and F scores. They also demonstrate the highest macro AUC above 0.98. XGBoost demonstrates a slightly better performance as AUC for XGBoost is 0.986 compared to 0.984 for Random Forest. Hence, XGBoost is the model to be used for new unseen data using the Distance feature.

The roc curves were also generated for the models that utilize Geodesic Distance. The plots are illustrated below.

A graph of a curve

Description automatically generated

A graph of a curve

Description automatically generated

A graph of a curve

Description automatically generatedA graph of a curve

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models with Geodesic Distance | **Decision Tree** | **Random Forest** | **XGBoost** | **Stacked Model** |
| **Precision (Average)** | 0.84 | **0.91** | 0.89 | 0.89 |
| **Recall (Average)** | 0.84 | **0.91** | 0.89 | 0.88 |
| **F-Score (Average)** | 0.84 | **0.91** | 0.89 | 0.88 |
| **AUC (Average)** | 0.931 | **0.9786** | 0.9784 | 0.937 |
| **Accuracy** | 0.84 | **0.91** | 0.90 | 0.88 |

From the table above it is evident that XGBoost and Random Forest demonstrate the best overall performance for Precision, Recall and F scores. They also demonstrate the highest macro AUC above 0.97. Random Forest demonstrates a slightly better performance as AUC for is 0.9786 compared to 0.9784 for XGBoost. Hence, Random Forest is the model to be used for new unseen data using the Geodesic Distance feature.

# References

Ababio-Donkor, A., Saleh, W. & Fonzone, A. 2020, "Understanding transport mode choice for commuting: the role of affect", *Transportation planning and technology,*vol. 43, no. 4, pp. 385-403.

Amiri, I.S., Akanbi, O.A. and Fazeldehkordi, E. (2014) *A Machine-Learning Approach to Phishing Detection and Defense*. 1st edn. Rockland, MA: Elsevier Science.

Amor, N.B., Benferhat, S. and Elouedi, Z. (2006) “Qualitative Classification with Possibilistic Decision Trees,” in *Modern Information Processing*. Elsevier B.V, pp. 159–169.

Bei, H. et al. (2023) “Joint prediction of travel mode choice and purpose from travel surveys: A multitask deep learning approach,” Travel, behaviour & society, 33, p. 100625.

Bellini, T. (2019) *IFRS 9 and CECL credit risk modelling and validation a practical guide with examples worked in R and SAS*. London: Academic Press.

Bhuiya, M.M.R. et al. (2022) “Application of Machine Learning Classifiers for Mode Choice Modeling for Movement-Challenged Persons,” Future transportation, 2(2), pp. 328–346.

Böcker, L., Dijst, M. and Faber, J. (2016) “Weather, transport mode choices and emotional travel experiences,” Transportation research. Part A, Policy and practice, 94, pp. 360–373.

Brownlee, J. (2020) *ROC curves and precision-recall curves for imbalanced classification*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/ (Accessed: 20 February 2024).

Brownlee, J. (2021) Random oversampling and undersampling for imbalanced classification, MachineLearningMastery.com. Available at: https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/ (Accessed: 12 January 2024).

Brownlee, J. (2021) *Stacking Ensemble Machine Learning with python*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/ (Accessed: 20 February 2024).

Cunningham, P. and Delany, S.J. (2021) “k-Nearest Neighbour Classifiers - A Tutorial,” *ACM computing surveys*, 54(6), pp. 1–25.

Das, S. *et al.* (2021) “Impact of COVID-19: A radical modal shift from public to private transport mode,” *Transport policy*, 109, pp. 1–11.

Hagenauer, J. and Helbich, M. (2017) “A comparative study of machine learning classifiers for modeling travel mode choice,” Expert systems with applications, 78, pp. 273–282.

Kettle, S. (2014) Geodesic distances: How long is that line again?, Esri Community. Available at: https://community.esri.com/t5/coordinate-reference-systems-blog/geodesic-distances-how-long-is-that-line-again/ba-p/902188 (Accessed: 25 January 2024).

Kubat, M. (2021) An Introduction to Machine Learning. 3rd edn. Cham: Springer International Publishing AG.

Kufel, J. et al. (2023) “What Is Machine Learning, Artificial Neural Networks and Deep Learning?-Examples of Practical Applications in Medicine,” Diagnostics (Basel), 13(15), p. 2582.

Lee, E. and Kim, S. (2023) *Geographic Information Systems for Intermodal Transportation*. Elsevier.

Lopez, F. (2021) Ensemble learning: Bagging & boosting, Medium. Available at: https://towardsdatascience.com/ensemble-learning-bagging-boosting-3098079e5422 (Accessed: 12 January 2024).

Mayo, F.L. & Taboada, E.B. 2020, "Ranking factors affecting public transport mode choice of commuters in an urban city of a developing country using analytic hierarchy process: The case of Metro Cebu, Philippines", *Transportation research interdisciplinary perspectives,*vol. 4, pp. 100078.

McCarthy, L., Delbosc, A., Currie, G. & Molloy, A. 2017, "Factors influencing travel mode choice among families with young children (aged 0-4): a review of the literature", *Transport reviews,*vol. 37, no. 6, pp. 767-781.

McFadden, D. (1972). Conditional logit analysis of qualitative choice behavior.

McFadden, D. (1974) “The measurement of urban travel demand,” *Journal of public economics*, 3(4), pp. 303–328.

McFadden, D. and Train, K. (2000) “Mixed MNL models for discrete response,” *Journal of applied econometrics (Chichester, England)*, 15(5), pp. 447–470.

Misra, S., Li, H. & He, J. 2019, Machine Learning for Subsurface Characterization, 1st edn, Elsevier Science & Technology, San Diego.

*Moped* (2024) *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Moped (Accessed: 30 January 2024).

Mussone, L. & Changizi, F. 2023, "A study on the factors that influenced the choice of transport mode before, during, and after the first lockdown in Milan, Italy", *Cities,*vol. 136, pp. 104251-104251.

Narkhede, S. (2018) *Understanding AUC - roc curve*, *Medium*. Available at: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (Accessed: 20 February 2024).

Omrani, H. (2015) ‘Predicting travel mode of individuals by machine learning’, *Transportation Research Procedia*, 10, pp. 840–849. doi:10.1016/j.trpro.2015.09.037.

Polamuri (2023) 7 most popular boosting algorithms to improve machine learning model’s performance, Dataaspirant. Available at: https://dataaspirant.com/boosting-algorithms/ (Accessed: 12 January 2024).

Raschka, S. (2023) *Stackingcvclassifier: Stacking with cross-validation*, *StackingCVClassifier: Stacking with cross-validation - mlxtend*. Available at: https://rasbt.github.io/mlxtend/user\_guide/classifier/StackingCVClassifier/ (Accessed: 20 February 2024).

Rocca, J. (2019) Ensemble methods: Bagging, boosting and stacking, Medium. Available at: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205 (Accessed: 11 January 2024).

Shanmugam, L. and Ramasamy, M. (2021) “Study on mode choice using nested logit models in travel towards Chennai metropolitan city,” Journal of ambient intelligence and humanized computing, pp. Journal of ambient intelligence and humanized computing, 2021.

Souza, G.F.M.de *et al.* (2021) *Reliability Analysis and Asset Management of Engineering Systems*. 1st edn. San Diego: Elsevier (Advances in Reliability Science).

Statistics Netherlands (2024) *Dutch National Travel Survey*, *Statistics Netherlands*. Available at: https://www.cbs.nl/en-gb/onze-diensten/methods/surveys/korte-onderzoeksbeschrijvingen/dutch-national-travel-survey (Accessed: 29 January 2024).

Stefan Trueck, S.T.R. (2009) Rating Based Modeling of Credit Risk. 1st edn. San Diego: Elsevier Science.

Tamim Kashifi, M. et al. (2022) “Predicting the travel mode choice with interpretable machine learning techniques: A comparative study,” Travel, behaviour & society, 29, pp. 279–296.

Wang, S., Mo, B. and Zhao, J. (2021) “Theory-based residual neural networks: A synergy of discrete choice models and deep neural networks,” *Transportation research. Part B: methodological*, 146, pp. 333–358.

Willis, K.G. (2014) “The Use of Stated Preference Methods to Value Cultural Heritage,” *Handbook of the Economics of Art and Culture*, 2, pp. 145–181.

XGBoost (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/xgboost/ (Accessed: 12 January 2024).

Yang, C., Fridgeirsson, E.A., Kors, J.A., Reps, J.M. & Rijnbeek, P.R. 2024, "Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data", Journal of big data, vol. 11, no. 1, pp. 1-17.

Zhang, Z. et al. (2020) “A Customized Deep Neural Network Approach to Investigate Travel Mode Choice with Interpretable Utility Information,” Journal of advanced transportation, 2020, pp. 1–11.

Zhao, X. et al. (2020) “Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models,” Travel, behaviour & society, 20(C), pp. 22–35.

Zhou, ZH. (2021). Introduction. In: Machine Learning. Springer, Singapore. https://doi.org/10.1007/978-981-15-1967-3\_1